



UNIVERSIDAD CARLOS III DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

INGENIERÍA INFORMÁTICA

PROYECTO FIN DE CARRERA

**Análisis de algoritmos basados en
colonia de hormigas en problemas de
camino mínimo**



UNIVERSIDAD CARLOS III DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

INGENIERÍA INFORMÁTICA

PROYECTO FIN DE CARRERA

**Análisis de algoritmos basados en
colonia de hormigas en problemas de
camino mínimo**

Directores: Pedro Isasi Viñuela
David Quintana Montero

Autor: Jesús Rodríguez García

Enero, 2010

Este proyecto se lo dedico a todas las personas que han confiado en mí, y en los momentos difíciles, su apoyo me ha servido para seguir avanzando y mantener la ilusión.

A mi familia por aguantar mis ausencias y siempre tenerles animándome.

A Pedro y David por haberme permitido la realización del presente proyecto, aportando valiosas sugerencias y correcciones.

Índice



ÍNDICE DE CONTENIDOS

1	INTRODUCCIÓN.....	2
1.1	CONTEXTO.....	2
1.2	OBJETIVOS.....	3
1.3	ESTRUCTURA.....	3
2	ESTADO DEL ARTE.....	5
2.1	GENERALIDADES SOBRE GRAFOS.....	5
2.1.1	<i>Definiciones básicas.....</i>	5
2.1.2	<i>Representación de un grafo.....</i>	7
2.1.2.1	Matriz de adyacencias.....	7
2.1.2.2	Matriz de incidencias.....	7
2.1.2.3	Lista de adyacencias.....	8
2.1.2.4	Lista de incidencias.....	8
2.2	TÉCNICAS TRADICIONALES DE BUSQUEDA EN GRAFOS.....	9
2.2.1	<i>Busqueda sin información.....</i>	9
2.2.1.1	Busqueda en amplitud.....	9
2.2.1.1	Busqueda en profundidad.....	10
2.2.2	<i>Grafos Ponderados.....</i>	11
2.2.2.1	Algoritmo de Dijkstra.....	11
2.2.3	<i>Busqueda heurística.....</i>	15
2.2.3.1	Busqueda en escalada.....	15
2.2.3.2	Busqueda primero el mejor.....	16
2.2.3.3	Algoritmo A*.....	17
2.2.3.4	Estrategia Minimax.....	19
2.2.3.5	Minimax con Poda. Metodo de poda α - β	20
2.3	SOLUCIONES AL PROBLEMA BASADAS EN METAHEURÍSTICAS.....	21
2.3.1	<i>Metaheurística basada en colonias de hormigas.....</i>	23
2.3.2	<i>Colonias de hormigas naturales.....</i>	24
2.3.2.1	Depósito de Feromonas.....	25
2.3.2.2	Sistema contador de pasos.....	28
2.3.3	<i>Colonias de hormigas artificiales.....</i>	29
2.3.4	<i>Modelos de optimización basados en colonias de hormigas.....</i>	30
2.3.4.1	Ant System. Algoritmo básico.....	30
2.3.4.2	Ant System y Elitismo.....	37
2.3.4.3	Ant Colony System (ACS).....	37
2.3.4.4	Max-Min Ant System. (MMAS).....	39
2.3.4.5	Sistema mejor-peor hormiga. (SMPH).....	41
2.3.4.6	Pararelización.....	42
2.3.5	<i>Problemas combinatoriales tratados.....</i>	43
3	RECORRIDOS DE GRAFOS MEDIANTE HORMIGAS.....	46
3.1	OBJETIVOS.....	47
3.2	ALGORITMO ANT SYSTEM EN EL CÁLCULO DE RUTAS DE DISTANCIA MÍNIMA ENTRE DOS PUNTOS CUALESQUIERA DE UNA RED.....	48
3.2.1	<i>Análisis del sistema.....</i>	48
3.2.2	<i>Diseño del sistema e implementación.....</i>	50
3.2.2.1	Diseño de la base de datos.....	50
3.2.2.2	Diseño del software.....	52
3.2.3	<i>Resultados.....</i>	64
3.2.4	<i>Problemas encontrados.....</i>	68
3.3	DIVISIÓN DEL OBJETIVO EN ALGORITMOS BASADOS EN COLONIAS DE HORMIGAS PARA EL CÁLCULO DE DISTANCIAS MÍNIMAS.....	69
3.3.1	<i>Modificaciones planteadas.....</i>	69
3.3.2	<i>Modificaciones realizadas.....</i>	70
3.3.2.1	Diseño de arcos y puntos.....	70
3.3.2.2	Modificaciones realizadas en el algoritmo.....	71
3.3.3	<i>Resultados.....</i>	75



UAnálisis de algoritmos basados en colonia de hormigas en problemas de camino mínimo

3.3.4	<i>Problemas encontrados.</i>	77
3.4	ORIENTACIÓN DE LA COLONIA DE HORMIGAS EN EL DESCUBRIMIENTO DEL DESTINO.	80
3.4.1	<i>Modificaciones planteadas.</i>	81
3.4.2	<i>Modificaciones realizadas.</i>	81
3.4.2.1	Modificación del algoritmo.	82
3.4.3	<i>Resultados.</i>	83
3.4.4	<i>Problemas encontrados.</i>	84
3.5	ESTUDIO SOBRE LOS DEPÓSITOS DE FEROMANA EN BASE A UNA REFERENCIA ÚNICA.	88
3.5.1	<i>Modificaciones planteadas.</i>	89
3.5.2	<i>Modificaciones realizadas.</i>	91
3.5.2.1	Modificación de la estructura de datos.	91
3.5.2.2	Modificación del algoritmo.	92
3.5.3	<i>Refinamiento de la solución adoptada.</i>	101
3.5.3.1	Centrar a la colonia de hormigas en el problema.	101
3.5.3.2	Valoración de feromona para la toma de decisión	102
3.5.3.3	Aporte de feromona de la hormiga fiel	104
3.5.3.4	Eliminación de las hormigas consideradas perdidas.	104
3.5.4	<i>Resultados.</i>	105
3.5.4.1	Análisis número de hormigas de la colonia.	106
3.5.4.2	Análisis porcentaje en origen de las hormigas	112
3.5.4.3	Análisis factor de feromona inicial.	115
3.5.4.4	Análisis de dimensiones de red.	119
3.5.4.5	Evolución de los resultados.	123
3.5.5	<i>Verificación del avance en la obtención de resultados por la introducción de mejoras en el algoritmo tomado como base.</i>	125
4	CONCLUSIONES Y MEJORAS.	127
5	BIBLIOGRAFÍA Y REFERENCIAS	131
	ANEXOS	137
A	GUÍA DE INSTALACIÓN Y EJECUCIÓN.	139
A.1	INSTALACIÓN DEL SOFTWARE ACO-UC3M.	139
A.2	GUÍA DE USUARIO.	142
A.2.1	<i>Gestión de red.</i>	143
A.2.2	<i>Variables del problema.</i>	144
A.2.3	<i>Ejecución y resultados.</i>	145
A.2.4	<i>Ficheros de salida.</i>	146
B	INFORMACIÓN DE SALIDA.	151
B.1	RESULTADOS. ARCHIVOS LOG	151
B.1.1	<i>Mejores hormigas de la colonia.</i>	151
B.1.2	<i>Mejores hormigas fiel.</i>	152
B.1.3	<i>Resultados.</i>	153
B.1.4	<i>Matriz de feromonas</i>	154
C	ESTRUCTURA DEL CD ANEXO.	157
C.1	DOCUMENTACIÓN.	158
C.1.1	<i>Documentación de referencia.</i>	158
C.1.2	<i>Documentación generada.</i>	158
C.2	FICHEROS DE SALIDA. RESULTADOS.	159
C.2.1	<i>Resultados - Análisis comportamiento del número de hormigas.</i>	159
C.2.2	<i>Resultados - Análisis de feromona de inicio.</i>	159
C.2.3	<i>Resultados - Análisis de hormigas en punto origen.</i>	160
C.2.4	<i>Resultados - Evolución de éxitos.</i>	160
C.3	SOFTWARE DESARROLLADO.	160
C.3.1	<i>Código fuente.</i>	160
C.3.2	<i>Instalación.</i>	161



ÍNDICE DE TABLAS

TABLA 1 – ALGORITMO DE DIJKSTRA. EJEMPLO.....	14
TABLA 2 – APORTE FEROMONA. EXPERIMENTO DE SIMON GOSS.....	27
TABLA 3 - ALGORITMO ANT SYSTEM.	32
TABLA 4 – ELECCIÓN DE MOVIMIENTO.	36
TABLA 5 - DISEÑO BASE DE DATOS. TABLA PUNTOS	51
TABLA 6 - DISEÑO BASE DE DATOS. TABLA ARCOS.	51
TABLA 7 – CLASE CPUNTO. PROPIEDADES.	55
TABLA 8 – CLASE CPUNTO. FUNCIONES.....	55
TABLA 9 – CLASE CARCO. PROPIEDADES.....	56
TABLA 10 – CLASE CARCO. FUNCIONES.	56
TABLA 11 – CLASE CHORMIGA. PROPIEDADES	58
TABLA 12 – CLASE CHORMIGA FUNCIONES.	58
TABLA 13 – CLASE CLISTAHORMIGA. PROPIEDADES.....	59
TABLA 14 – CLASE CLISTAHORMIGA. FUNCIONES.	59
TABLA 15 – CLASE CLISTAPUNTOS. PROPIEDADES.....	60
TABLA 16 – CLASE CLISTAPUNTOS. FUNCIONES.	60
TABLA 17 – ENTORNO EXPERIMENTAL. I_R100-2-50.....	66
TABLA 18 – ENTORNO EXPERIMENTAL. I_R100-50-50.	66
TABLA 19 – ENTORNO EXPERIMENTAL. I_R100-100-50	67
TABLA 19 – EVOLUCIÓN PUNTOS. CLASIFICACIÓN.	73
TABLA 20 – ENTORNO EXPERIMENTAL. II_R100-2-50.....	75
TABLA 21 – ENTORNO EXPERIMENTAL. I_R1000-20-50.....	80
TABLA 22 – ENTORNO EXPERIMENTAL. III_R100-2-50	84
TABLA 23 – ERROR DEPOSICIÓN DE FEROMONA. PROBLEMAS ENCONTRADOS.....	86
TABLA 24 – ERROR FEROMONA INICIAL. PROBLEMAS ENCONTRADOS.....	87
TABLA 25 – SIMULACIÓN DEL COMPORTAMIENTO DE LA HORMIGA CATAGLYPHIS FORTIS.	89
TABLA 26 – EVOLUCIÓN DEL VALOR DE FEROMONA.	98
TABLA 27 – COMPORTAMIENTO DE LA HORMIGA FIEL.....	99
TABLA 28 – VENTAJAS DE LA HORMIGA FIEL.....	101
TABLA 29 – ENTORNO EXPERIMENTAL. CÁLCULO NÚMERO HORMIGAS. IV_R500-NH.....	106
TABLA 30 – ENTORNO EXPERIMENTAL. CÁLCULO NÚMERO HORMIGAS. IV_R1000-NH.....	108
TABLA 31 – ENTORNO EXPERIMENTAL. CÁLCULO NÚMERO HORMIGAS. IV_R2000-NH.....	109
TABLA 32 – ENTORNO EXPERIMENTAL. CÁLCULO NÚMERO HORMIGAS. IV_R5000-NH.....	110
TABLA 33 – ENTORNO EXPERIMENTAL. CÁLCULO PORCENTAJE EN ORIGEN. IV_R5000-PO.	112
TABLA 34 – ENTORNO EXPERIMENTAL. CÁLCULO DEL FACTOR DE FEROMONA INICIAL. IV_R1000-FI....	116
TABLA 35 – ENTORNO EXPERIMENTAL. ANÁLISIS DE DIMENSIONAMIENTO DE LA RED. IV_R100-DR.	119
TABLA 36 – ENTORNO EXPERIMENTAL. ANÁLISIS DE DIMENSIONAMIENTO DE LA RED. IV_R500-DR.	120
TABLA 37 – ENTORNO EXPERIMENTAL. ANÁLISIS DE DIMENSIONAMIENTO DE LA RED. IV_R1000-DR. ..	121
TABLA 38 – ENTORNO EXPERIMENTAL. ANÁLISIS DE DIMENSIONAMIENTO DE LA RED. IV_R2000-DR. ..	122
TABLA 39 – ENTORNO EXPERIMENTAL. EVOLUCIÓN DE LOS RESULTADOS. IV_R500-1000-E.	123



ÍNDICE DE ILUSTRACIONES

ILUSTRACIÓN 1 – GRAFO DIRIGIDO.....	5
ILUSTRACIÓN 2 – GRAFO NO DIRIGIDO.....	6
ILUSTRACIÓN 3 – REPRESENTACIÓN DE UN GRAFO. MATRIZ DE ADYACENCIAS.....	7
ILUSTRACIÓN 4 – REPRESENTACIÓN DE UN GRAFO. MATRIZ DE INCIDENCIAS.....	8
ILUSTRACIÓN 5 – REPRESENTACIÓN DE UN GRAFO. LISTA DE ADYACENCIA.....	8
ILUSTRACIÓN 6 – REPRESENTACIÓN DE UN GRAFO. LISTA DE INCIDENCIAS.....	9
ILUSTRACIÓN 7 – RECORRIDO EN AMPLITUD.....	9
ILUSTRACIÓN 8 – RECORRIDO EN PROFUNDIDAD.....	10
ILUSTRACIÓN 9 – COMPORTAMIENTO DE LA HORMIGA. DECISIÓN DE RUTA -INICIAL-.....	26
ILUSTRACIÓN 10 – COMPORTAMIENTO DE LA HORMIGA. DECISIÓN DE RUTA -FINAL-.....	27
ILUSTRACIÓN 11 – CATAGLYPHIS FORTIS. ZANCOS DE MODIFICACIÓN DE AMPLITUD DE PASO.....	28
ILUSTRACIÓN 12 – PROCESO DE ELABORACIÓN DE CADA FASE DEL PROYECTO.....	46
ILUSTRACIÓN 13 – DISEÑO DE LA RED DE PUNTOS DE PRUEBAS.....	49
ILUSTRACIÓN 14 – MODELO DE DATOS ESTÁTICO.....	51
ILUSTRACIÓN 15 – PRINCIPALES FUNCIONALIDADES.....	52
ILUSTRACIÓN 16 – PRINCIPALES FUNCIONALIDADES.....	53
ILUSTRACIÓN 17 – ESTRUCTURA DE REPRESENTACIÓN. PUNTO.....	54
ILUSTRACIÓN 18 – ESTRUCTURA DE REPRESENTACIÓN. ARCO.....	56
ILUSTRACIÓN 19 – ESTRUCTURA DE REPRESENTACIÓN. PUNTO Y ARCOS.....	57
ILUSTRACIÓN 20 – ESTRUCTURA DE REPRESENTACIÓN. HORMIGA.....	57
ILUSTRACIÓN 21 – ESTRUCTURA DE REPRESENTACIÓN. COLONIA DE HORMIGAS.....	59
ILUSTRACIÓN 22 – HEURÍSTICA EN EL ALGORITMO PLANTEADO.....	61
ILUSTRACIÓN 23 – REPRESENTACIÓN GRÁFICA DEL ALGORITMO.....	62
ILUSTRACIÓN 24 – RESULTADOS. I_R100-2-50.....	66
ILUSTRACIÓN 25 – RESULTADOS. I_R100-50-50.....	66
ILUSTRACIÓN 26 – RESULTADOS. I_R100-100-50.....	67
ILUSTRACIÓN 27 – REPRESENTACIÓN DE PUNTO. MODELO BIDIRECCIONAL.....	70
ILUSTRACIÓN 28 – DISEÑO ARCOS.....	71
ILUSTRACIÓN 30 – ALGORITMO ACO-DIVISIÓN DEL PROBLEMA.....	75
ILUSTRACIÓN 31 – RESULTADOS. MODELO II_R100-2-50.....	76
ILUSTRACIÓN 32 – COMPORTAMIENTO ALGORITMO.....	77
ILUSTRACIÓN 33 – POSIBLE SOLUCIÓN AL MODELO PLANTEADO.....	78
ILUSTRACIÓN 34 – POSIBLE SOLUCIÓN AL MODELO PLANTEADO II.....	79
ILUSTRACIÓN 35 – RESULTADOS I_R1000-20-50. COMPORTAMIENTO ALGORITMO. 1 ^{ER} ÉXITO.....	80
ILUSTRACIÓN 36 – ESTRUCTURA HORMIGA.....	82
ILUSTRACIÓN 37 – ALGORITMO ACO-ORIENTACIÓN DE LA COLONIA DE HORMIGAS.....	83
ILUSTRACIÓN 38 – RESULTADOS. III_R100-2-50.....	84
ILUSTRACIÓN 39 – VALORES INICIALES DE FEROMONA.....	90
ILUSTRACIÓN 40 – ESTRUCTURA DE DATOS DEL ELEMENTO ARCO.....	91
ILUSTRACIÓN 41 – ESTRUCTURA DE DATOS DEL ELEMENTO PUNTO.....	91
ILUSTRACIÓN 42 – ALGORITMO ACO-REFERENCIA ÚNICA.....	92
ILUSTRACIÓN 43 – ALGORITMO DE DEPOSICIÓN DE FEROMONA.....	94
ILUSTRACIÓN 44 – ZONA DE MAYOR PROBABILIDAD DE VISITA.....	102
ILUSTRACIÓN 45 – FACTOR DE VALORACIÓN DE LA FEROMONA INICIAL.....	102
ILUSTRACIÓN 46 – FACTOR DE FEROMONA INICIAL. VALOR DE K NEUTRO.....	103
ILUSTRACIÓN 47 – FACTOR DE FEROMONA INICIAL. VALORES DE K MENORES A UNO.....	103
ILUSTRACIÓN 48 – FACTOR DE FEROMONA INICIAL. VALORES DE K SUPERIORES A UNO.....	104
ILUSTRACIÓN 49 – VALORACIÓN DE HORMIGAS PERDIDAS.....	105
ILUSTRACIÓN 50 – RESULTADOS. CÁLCULO NÚMERO HORMIGAS. IV_R500-NH.....	106
ILUSTRACIÓN 51 – GRÁFICO. CÁLCULO NÚMERO HORMIGAS. IV_R500-NH.....	107
ILUSTRACIÓN 52 – RESULTADOS. CÁLCULO NÚMERO HORMIGAS. IV_R1000-NH.....	108
ILUSTRACIÓN 53 – GRÁFICO. CÁLCULO NÚMERO HORMIGAS. IV_R1000-NH.....	108
ILUSTRACIÓN 54 – RESULTADOS. CÁLCULO NÚMERO HORMIGAS. IV_R2000-NH.....	109
ILUSTRACIÓN 55 – GRÁFICO. CÁLCULO NÚMERO HORMIGAS. IV_R2000-NH.....	110



**UAnálisis de algoritmos basados en colonia de hormigas en problemas de
camino mínimo**

ILUSTRACIÓN 56 – RESULTADOS. CÁLCULO NÚMERO HORMIGAS. IV_R5000-NH.	111
ILUSTRACIÓN 57 – GRÁFICO. CÁLCULO NÚMERO HORMIGAS. IV_R5000-NH.	111
ILUSTRACIÓN 58 – RESULTADOS. CÁLCULO PORCENTAJE EN ORIGEN. IV_R5000-PO-I.	112
ILUSTRACIÓN 59 – RESULTADOS. CÁLCULO PORCENTAJE EN ORIGEN. IV_R5000-PO-II.	113
ILUSTRACIÓN 60 – GRÁFICO. CÁLCULO PORCENTAJE EN ORIGEN. IV_R5000-PO-HC.	113
ILUSTRACIÓN 61 – GRÁFICO. CÁLCULO PORCENTAJE EN ORIGEN. IV_R5000-PO-HF.	114
ILUSTRACIÓN 62 – GRÁFICO. CÁLCULO PORCENTAJE EN ORIGEN. IV_R5000-PO-HCT.	114
ILUSTRACIÓN 63 – GRÁFICO. CÁLCULO PORCENTAJE EN ORIGEN. IV_R5000-PO-HFT.	115
ILUSTRACIÓN 64 – RESULTADOS. CÁLCULO DEL FACTOR DE FEROMONA INICIAL. IV_R1000-FI-I.	116
ILUSTRACIÓN 65 – RESULTADOS. CÁLCULO DEL FACTOR DE FEROMONA INICIAL. IV_R1000-FI-II.	116
ILUSTRACIÓN 66 – GRÁFICO. CÁLCULO DEL FACTOR DE FEROMONA INICIAL. IV_R1000-FI. CICLO HORMIGA COLONIA.	117
ILUSTRACIÓN 67 – GRÁFICO. CÁLCULO DEL FACTOR DE FEROMONA INICIAL. IV_R1000-FI. CICLO HORMIGA FIEL.	117
ILUSTRACIÓN 68 – GRÁFICO. CÁLCULO DEL FACTOR DE FEROMONA INICIAL. IV_R1000-FI. TIEMPO HORMIGA COLONIA.	118
ILUSTRACIÓN 69 – GRÁFICO. CÁLCULO DEL FACTOR DE FEROMONA INICIAL. IV_R1000-FI. TIEMPO HORMIGA FIEL.	118
ILUSTRACIÓN 70 – RESULTADOS. ANÁLISIS DE DIMENSIONAMIENTO DE LA RED. IV_R100-DR.	120
ILUSTRACIÓN 71 – RESULTADOS. ANÁLISIS DE DIMENSIONAMIENTO DE LA RED. IV_R500-DR.	121
ILUSTRACIÓN 72 – RESULTADOS. ANÁLISIS DE DIMENSIONAMIENTO DE LA RED. IV_R1000-DR.	122
ILUSTRACIÓN 73 – RESULTADOS. ANÁLISIS DE DIMENSIONAMIENTO DE LA RED. IV_R2000-DR.	123
ILUSTRACIÓN 74 – RESULTADOS. EVOLUCIÓN DE LOS RESULTADOS. IV_R500-1000-E.	124
ILUSTRACIÓN 75 – GRÁFICO. ANÁLISIS DE EVOLUCIÓN DE LOS RESULTADOS. IV_R500-E.	124
ILUSTRACIÓN 76 – GRÁFICO. ANÁLISIS DE EVOLUCIÓN DE LOS RESULTADOS. IV_R1000-E.	125
ILUSTRACIÓN 77 – PANTALLA DE INSTALACIÓN I.	139
ILUSTRACIÓN 78 – PANTALLA DE INSTALACIÓN II.	140
ILUSTRACIÓN 79 – PANTALLA DE INSTALACIÓN III.	140
ILUSTRACIÓN 80 – PANTALLA DE INSTALACIÓN IV.	141
ILUSTRACIÓN 81 – PROGRAMA ACO-UC3M- PANTALLA DE INICIO.	142
ILUSTRACIÓN 82 – PROGRAMA ACO-UC3M- GESTIÓN DE RED.	143
ILUSTRACIÓN 83 – PROGRAMA ACO-UC3M- VARIABLES DEL PROBLEMA.	144
ILUSTRACIÓN 84 – PROGRAMA ACO-UC3M. EJECUCIÓN-ACO-RESULTADOS.	145
ILUSTRACIÓN 85 – PROGRAMA ACO-UC3M- FICHEROS DE SALIDA.	147
ILUSTRACIÓN 86 – ESTRUCTURA CD ANEXO.	157
ILUSTRACIÓN 87 – ESTRUCTURA CD ANEXO. DOCUMENTACIÓN.	158
ILUSTRACIÓN 88 – ESTRUCTURA CD ANEXO. FICHEROS DE SALIDA - RESULTADOS.	159
ILUSTRACIÓN 89 – ESTRUCTURA CD ANEXO. SOFTWARE DESARROLLADO.	160

Memoria





1 INTRODUCCIÓN.

1.1 CONTEXTO.

Existe un conjunto de problemas donde la búsqueda de la solución óptima puede ser una tarea computacionalmente costosa. La resolución de estos mediante algoritmos deterministas sólo puede garantizarse en un tiempo de ejecución que crece exponencialmente con el tamaño del problema. Al no existir alternativas que los resuelvan en un tiempo polinomial [ALO00], las posibilidades de obtener resultados en tiempos aceptables se reducen dramáticamente para instancias grandes.

Esta complejidad es la que ha originado numerosos estudios y el desarrollo de diferentes técnicas capaces de enfrentarse a este tipo de problemas, aportando soluciones más o menos eficientes. La investigación para la resolución de problemas de este tipo en los últimos tiempos apunta hacia técnicas metaheurísticas, vistas como formas de trabajo para la resolución de diversos problemas de optimización combinatoria.

Realmente no existe un acuerdo común aceptado para la definición de metaheurística [BLU03][COB05], pero se puede definir como *"un proceso iterativo de generación que guía a una heurística subordinada combinando de forma inteligente diferentes conceptos para la exploración del espacio de búsqueda y usando estrategias de aprendizaje para estructurar la información con objeto de encontrar eficientemente soluciones cercanas al óptimo"*[OSM05].

El presente trabajo es un estudio del comportamiento de una de las técnicas de más reciente aparición bajo inspiración biológica [BON99], más concretamente las que toman como base el comportamiento de las Colonias de Hormigas [DOR91][DOR961].

La naturaleza, con el paso de tiempo, ha sido capaz de resolver una gran cantidad de problemas que se puede aplicar a otros muchos campos de la ciencia. El poder aprovechar el comportamiento, y por lo tanto, el conocimiento adquirido en el transcurso de la evolución de las especies, utilizándolas como expertos, puede servir para la resolución de problemas como el tratado en este proyecto.

Los algoritmos basados en colonias de hormigas toman como base el comportamiento utilizado por las hormigas para la resolución del problema de obtención del camino mínimo entre el hormiguero y su fuente de alimentación, basándose en la iteración de elementos simples, como son las hormigas, que de manera conjunta son capaces de resolver el problema mediante una comunicación a través del medio, premiando los comportamientos que mejor resultado aporten a la comunidad.

Estos modelos se han utilizado, con resultados esperanzadores, en problemas de explosión combinatoria en campos muy diversos, como pueden ser, problemas de inversión financiera, diseño de circuitos, mantenimiento de sistemas, redes hidráulicas, redes informáticas, por citar algunos ejemplos.



1.2 OBJETIVOS.

El objetivo del proyecto es el estudio mediante algoritmos de optimización basados colonias de hormigas, para la resolución de un problema combinatorial como es la búsqueda del camino más corto entre dos puntos cualesquiera en una red, problema que llevado al mundo real conlleva un gran interés económico y social.

El estudio se centra en la implementación y comprobación de la eficiencia de estos algoritmos sobre el problema en cuestión. Partiendo de estudios realizados sobre problemas similares, se procederá a la adaptación al problema objeto de estudio, identificando las dificultades y errores que pudieran surgir e intentado solucionarlos siguiendo la filosofía de este tipo de algoritmos, buscando la mejora de los resultados y cuantificando el tamaño donde se presentan unas soluciones aceptables.

Las optimizaciones de rutas son un problema en empresas de muchos campos, y cualquier mejora, por pequeña que esta sea, puede suponer una mayor eficiencia y por lo tanto una mayor competitividad.

1.3 ESTRUCTURA.

El proyecto se divide en cinco capítulos, de ellos, este primero, presenta de manera general un enfoque del desarrollo del proyecto, así como sus causas y fines.

En el capítulo 2, se hace un breve repaso de la evolución y estado actual de los temas de estudio en el proyecto, exponiendo generalidades sobre la teoría de grafos, sus métodos tradicionales de búsqueda, así como los principales algoritmos basados en colonias de hormigas

En el capítulo 3, se presenta el trabajo realizado, marcando en un primer momento los objetivos que se quieren alcanzar, seguido de las diferentes fases desarrolladas con los problemas encontrados y como se han ido resolviendo para el cumplimiento de los objetivos.

Finalizado el estudio del problema y observando los resultados obtenidos, el capítulo 4, de manera muy concisa, informa de las impresiones más relevantes que se han obtenido de la elaboración del proyecto, así como futuros estudios que se sugieren a desarrollar.

Y para finalizar, el capítulo 5 informa de la bibliografía consultada para la elaboración del proyecto.

Además de los capítulos mencionados, se presenta a continuación un anexo donde se expone información adicional que se considera útil como complemento al proyecto.

2 ESTADO DEL ARTE.

2.1 GENERALIDADES SOBRE GRAFOS.

La forma de representación del problema a tratar en el proyecto se ajusta perfectamente a la teoría de grafos, es por ello por lo que hacemos una breve descripción de algunos conceptos elementales [INF05], que posteriormente se utilizarán a lo largo del trabajo.

En 1736 Leonhard Euler ya planteo el problema de los puentes de Königsberg, que es considerado el primer indicio en la teoría de grafos. Posteriormente ya en 1976, con la resolución del problema de los cuatro colores por parte de Kenneth Appel y Wolfgang Haken, fue cuando se ha considerado el nacimiento de la teoría de grafos [LUZ73]. Con la resolución del problema, se definieron los términos y conceptos teóricos fundamentales de la teoría de grafos.

El grafo es una estructura muy importante, tanto en Informática como Matemáticas. Muchos problemas de difícil resolución, pueden ser expresados en forma de grafo y por lo tanto resueltos usando algoritmos de búsqueda y manipulación estándar sobre grafos. Se puede considerar como una estructura no lineal, que representa un conjunto de nodos con arcos que conectan a estos.

2.1.1 DEFINICIONES BÁSICAS.

- **Grafo Dirigido**

Definición: es un par $G = (V;A)$ donde V es un conjunto finito de elementos llamados nodos y $A \subseteq V \times V$ es un conjunto de pares ordenados de nodos llamados arcos.

Si (u,v) es un arco de G , se dice que el nodo v es adyacente a u .

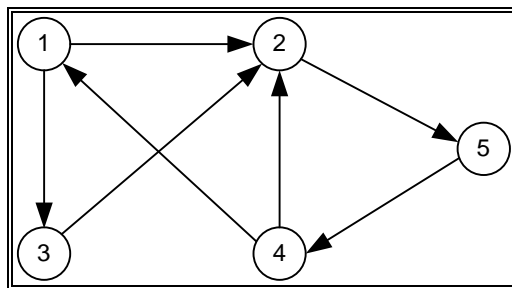


Ilustración 1 – Grafo dirigido.

- **Grafo no dirigido**

Definición: es un par $G = (V; A)$ donde V es un conjunto finito de nodos y $A \subseteq \{\{u, v\} | u, v \in V \wedge v \neq u\}$ es un conjunto de pares no ordenados de nodos.

Si $a = \{u, v\}$ es un arco no dirigido, se dice que a une a u y v y que a incide en u y v .

Si $\{u, v\}$ es un arco de G , se dice que el nodo v es adyacente a u . La relación es simétrica.

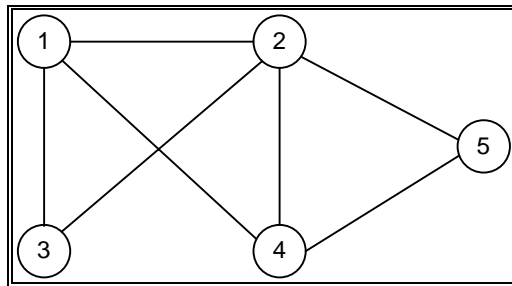


Ilustración 2 – Grafo no dirigido.

- **Grafo simple**

Es un grafo sin bucles en el que no hay dos aristas que unan el mismo par de vértices. Si un grafo no es simple se denomina **multigrafo**.

- **Grafo conexo:**

Si para cada par de nodos u y v existe una camino entre ellos.

- **Grado.**

Para todo nodo v :

Grado de entrada: Es el número de arcos que inciden en v .

Grado de salida: Es el número de arcos que parten de v .

Grado: Es la suma de los grados de entrada y salida.

- **Camino simple:**

Es aquel camino en el que todos sus nodos son distintos, excepto quizás el primero y el último.

- **Ciclo:**

Es el camino simple $\langle v_0, v_1, \dots, v_k \rangle$ donde el nodo inicial y final coinciden y contiene al menos un arco.

- **Longitud de un camino** $\langle v_0, v_1, \dots, v_k \rangle$ es el número de arcos que lo forman.

2.1.2 REPRESENTACIÓN DE UN GRAFO

Existen diferentes formas de poder representar un grafo. La estructura elegida dependerá de las características del grafo y del algoritmo utilizado para su manipulación.

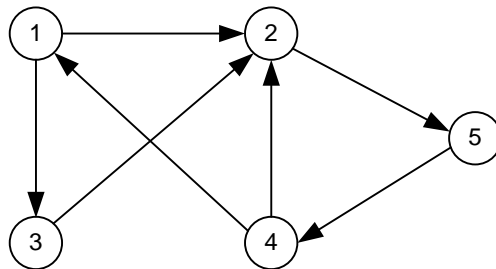
A continuación se indican las representaciones más utilizadas computacionalmente.

2.1.2.1 MATRIZ DE ADYACENCIAS

Sea G un grafo con n nodos $\{v_i\}_{i=1}^n$. Llamamos matriz de adyacencia a la matriz de orden $n \times n$, $A = [a_{ij}]$ tal que a_{ij} es igual al número de arcos del vértice v_i al v_j .

Esta representación presenta un acceso rápido, pero su consumo de memoria es elevado.

El grafo se representa mediante una matriz cuadrada de tamaño V^2 , donde V es el número de nodos.

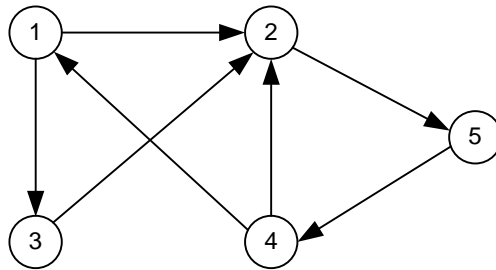


	1	2	3	4	5
1	0	1	1	0	0
2	0	0	0	0	1
3	0	1	0	0	0
4	1	1	0	0	0
5	0	0	0	1	0

Ilustración 3 – Representación de un grafo. Matriz de adyacencias.

2.1.2.2 MATRIZ DE INCIDENCIAS

Sea G un grafo con n nodos $\{v_i\}_{i=1}^n$. Llamamos matriz de incidencia a la matriz de orden $n \times n$, $A = [a_{ij}]$ tal que a_{ij} es igual al número de arcos del vértice v_j al v_i .



	1	2	3	4	5
1	0	0	0	1	0
2	1	0	1	1	0
3	1	0	0	0	0
4	0	0	0	0	1
5	0	1	0	0	0

Ilustración 4 – Representación de un grafo. Matriz de incidencias.

2.1.2.3 LISTA DE ADYACENCIAS

Se utiliza cuando los grafos son dispersos. Además proporciona una mejor utilización de la memoria del sistema.

Esta estructura, asocia a cada nodo del grafo una lista que contiene los arcos que se dirigen a los nodos que le son adyacentes, reservando memoria solamente para estos y no para todos los posibles arcos que pudiera tener como origen.

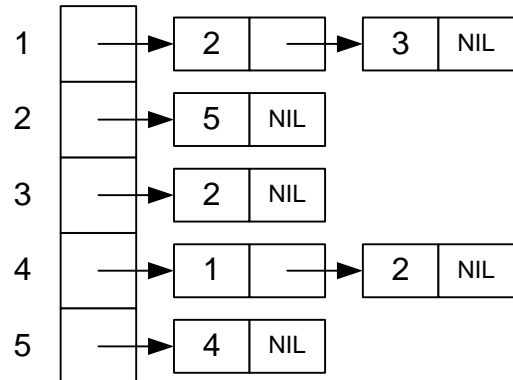
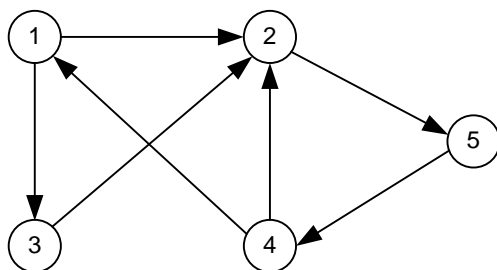


Ilustración 5 – Representación de un grafo. Lista de adyacencia.

2.1.2.4 LISTA DE INCIDENCIAS

Esta estructura, asocia a cada nodo del grafo una lista que contiene los arcos que se dirigen a los nodos que le son incidentes, reservando memoria solamente para estos y no para todos los posibles arcos que pudiera tener como destino.

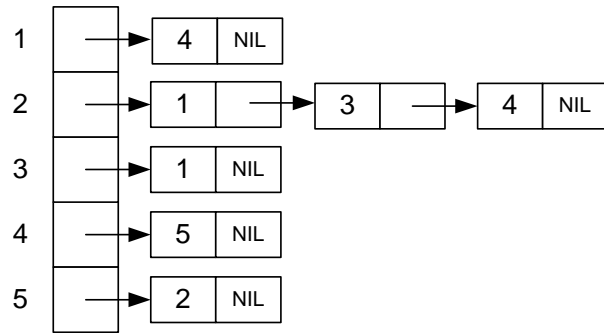
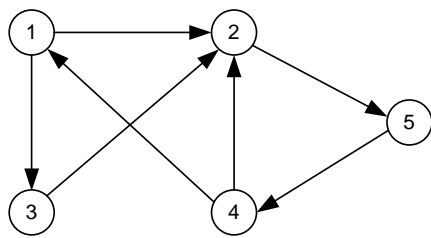


Ilustración 6 – Representación de un grafo. Lista de incidencias.

2.2 TÉCNICAS TRADICIONALES DE BUSQUEDA EN GRAFOS.

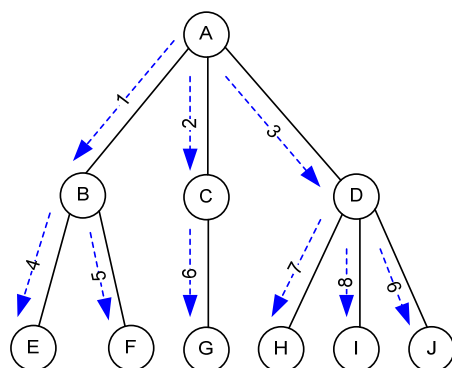
La mayoría de las acciones realizadas sobre grafos tienen que ver con su recorrido a partir de un determinado nodo. El problema de hallar un estado que satisfaga una condición puede formularse como la búsqueda de un grafo que posea un nodo cuya descripción asociada coincida con el estado objetivo. En realidad es una búsqueda y esta puede seguir dos patrones que a continuación se describen.

2.2.1 BÚSQUEDA SIN INFORMACIÓN.

Estudiaremos en primer lugar las formas básicas de control que permiten obtener, a través de un proceso exhaustivo, un estado meta. Consideramos aquéllas que realizan una búsqueda sistemática y objetiva. Se consideran objetivas por que el control del proceso de búsqueda no depende del problema concreto que se está resolviendo.

Antes de pasar al estudio de cada método conviene señalar que los principios que deben cumplir las estrategias sistemáticas de control son: no dejar –a priori- ningún nodo sin explorar y no explorar más de una vez el mismo nodo.

1.1.1.1 BÚSQUEDA EN AMPLITUD.



Recorrido : A-B-C-D-E-F-G-H-I-J

Ilustración 7 – Recorrido en amplitud.

La búsqueda en amplitud es uno de los algoritmos más simples para recorrer un grafo.

Dado un grafo G y un vértice de salida A, el algoritmo explora sistemáticamente los arcos de G para encontrar todos los nodos alcanzables desde A. Es decir, no se genera ningún nodo de nivel N hasta que no se hayan obtenido todos los del nivel N-1.

Ejemplo con arcos que presentan el mismo peso. Las acciones realizadas son las siguientes:

- Calcula la distancia de A a cada vértice alcanzable.
- Produce un árbol con raíz A que contiene todos los vértices alcanzables.
- Para cualquier nodo v alcanzable desde A , la ruta de A a v contiene la distancia mínima.
- **Ventajas:** Si el problema tiene solución este procedimiento garantiza que se encuentra y si hay varias metas encuentra la de menor coste (profundidad).
- **Desventajas:** Si el nivel de profundidad es menor que el nivel de ramificación se expanden muchos nodos inútilmente. Su principal desventaja es el espacio de almacenamiento requerido (prácticamente inviable).

La *complejidad temporal* de este método depende en gran medida del factor de ramificación y de la profundidad de la solución. Para un número medio de sucesores de n , alcanzando la solución en el nivel p , el tiempo empleado es $1 + n + n^2 + \dots + n^p$, siendo la complejidad del orden $O(n^p)$. En cuanto a la *complejidad espacial*, dado que antes de abandonar la generación de todos los sucesores de un nivel se deben de recordar todos los nodos de dicho nivel, la complejidad espacial es también del orden $O(n^p)$.

2.2.1.1 BÚSQUEDA EN PROFUNDIDAD.

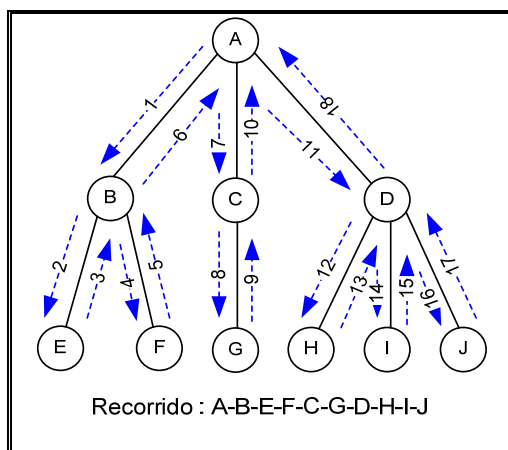


Ilustración 8 – Recorrido en profundidad.

El algoritmo en profundidad, como su nombre indica, busca más interiormente en el grafo mientras sea posible.

Explora sistemáticamente los arcos de manera que se visitan los nodos adyacentes a los visitados más recientemente. Esto origina una profundización en el grafo y por lo tanto un alejamiento del nodo inicial.

Cuando los arcos del nodo v han sido explorados, la búsqueda retrocede para explorar arcos que salen del nodo a partir del cual se descubrió v . Se continúa el proceso hasta que se identifican todos los nodos alcanzables desde el nodo de partida. Responde a un funcionamiento en forma de pila.

- **Ventajas:** Valor reducido de su complejidad espacial. Cuando hay múltiples soluciones la eficiencia del método aumenta.
- **Desventajas:** El tiempo requerido si el algoritmo avanza por una rama que no llevaría a solución, además si no se guarda constancia del camino recorrido se

puede caer en ciclos y el proceso no acabaría. El problema es pues determinar el límite de profundidad.

La *complejidad temporal* es la misma que en la búsqueda en amplitud, ya que para un grafo de una determinada profundidad p se generan los mismos nodos, aunque en diferente orden (las complejidades de las estrategias de búsqueda se calculan considerando el peor caso). En cuanto a la *complejidad espacial*, a lo largo del grafo de búsqueda sólo es necesario guardar constancia del camino construido hasta el momento, luego la complejidad para un camino de longitud p será dicho valor más el factor de ramificación (r), ya que cada vez es necesario guardar constancia de todos los sucesores del nodo que se está expandiendo, es decir $O(p + r)$.

Existen otros métodos de búsqueda, pero todos toman como base los anteriormente expuestos, entre ellos podemos destacar:

- **Búsqueda en profundidad progresiva.** Consiste en realizar una búsqueda en profundidad por niveles, de forma que el límite de exploración aumenta una unidad por cada nivel analizado. Esta estrategia es la más eficiente, dentro de las estrategias no informadas que son capaces de encontrar la solución de menor coste existente (óptima).
- **Búsqueda bidireccional.** Cuando además de la descripción del problema, se parte de una meta explícita, existe un planteamiento alternativo que resulta de la combinación de dos grafos de búsqueda diferentes. Se realiza simultáneamente la búsqueda de la meta (encaminamiento hacia adelante) y la búsqueda del estado inicial a partir de un estado solución (encaminamiento hacia atrás) hasta que ambos procesos confluyan en algún estado. Para garantizar la convergencia al menos uno de los procesos ha de ser en amplitud.

2.2.2 GRAFOS PONDERADOS.

Un grafo $G = (V, A)$ diremos que es ponderado si tiene asociado una función $W: A \rightarrow \mathbb{R}$ llamada función de ponderación. El valor de cada arco determinado por los nodos v_i y v_j le llamaremos peso del arco y lo denotaremos por w_{ij} .

En un grafo ponderado, llamamos peso de un camino, a la suma de los pesos de los arcos que lo forman.

Llamamos camino más corto entre dos nodos dados, al camino de peso mínimo entre dichos nodos.

Como camino critico entre dos nodos dados, entendemos al camino de peso máximo entre dichos nodos.

2.2.2.1 ALGORITMO DE DIJKSTRA

El algoritmo de Dijkstra identifica partiendo de un nodo de un grafo, el camino de coste mínimo entre este nodo y el resto de nodos que conforman el grafo.

Sea un grafo ponderado tal que $w_{ij} \geq 0$, este algoritmo encuentra los caminos más cortos, o de peso mínimo, desde el nodo origen al destino [VAL00].

La idea consiste en ir explorando todos los caminos más cercanos que parten del nodo origen y llevan al resto de nodos. El algoritmo finaliza cuando se ha calculado el camino más corto al resto de nodos o al nodo declarado como destino.

De una manera más formal se puede decir que el algoritmo mantiene un conjunto N de nodos a los que conocemos la distancia más corta desde u , junto con el predecesor de cada vértice de N en dicha ruta. Para lograr esto, mantenemos una distancia tentativa $t(z)$ desde u a cada $z \in N$. Si $z \notin N$, $t(z)$ es la distancia más corta encontrada hasta ahora entre u y z . Si $z \in N$, $t(z)$ es la distancia más corta entre u y z .

El algoritmo se puede describir en los siguientes pasos:

Teniendo un grafo dirigido ponderado de N nodos no aislados, sea x el nodo inicial, un vector D de tamaño N guardará al final del algoritmo las distancias desde x al resto de los nodos.

1. Inicializar todas las distancias en D con un valor infinito relativo ya que son desconocidas al principio, exceptuando la de x que se debe colocar en 0 debido a que la distancia de x a x sería 0.
2. Sea $a = x$ (tomamos a como nodo actual).
3. Recorremos todos los nodos adyacentes de a , excepto los nodos marcados, llamaremos a estos v_i .
4. Si la distancia desde x hasta v_i guardada en D es mayor que la distancia desde x hasta la sumada a la distancia desde a hasta v_i , esta se sustituye con la segunda nombrada, esto es: si $(D_i > D_a + d(a, v_i))$ entonces $D_i = D_a + d(a, v_i)$.
5. Marcamos como completo el nodo a .
6. Tomamos como próximo nodo actual el de menor valor en D (puede hacerse almacenando los valores en una cola de prioridad) y volvemos al paso 3 mientras existan nodos no marcados.

Una vez terminado al algoritmo, D estará completamente lleno.

Complejidad: En cada iteración se añade un vértice a T , luego el número de iteraciones es n . En cada una se elige una etiqueta mínima, la primera vez entre $n-1$, la segunda entre $n-2$, así sucesivamente, luego la complejidad total de estas elecciones es $O(n^2)$. Por otra parte cada arista da lugar a una actualización de etiqueta, que se puede hacer en tiempo constante $O(1)$, en total pues $O(q)$. Por tanto la complejidad total del algoritmo es $O(n^2)$.

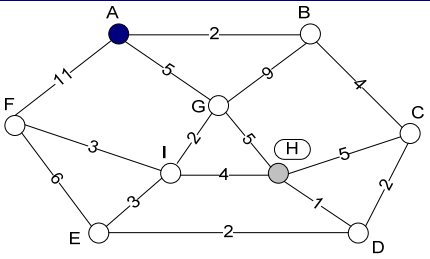
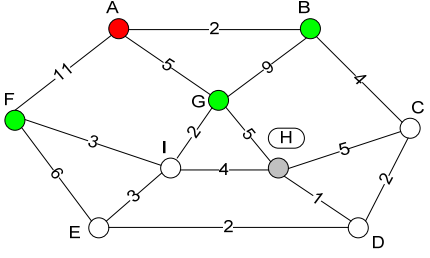
El pseudocódigo del algoritmo de Dijkstra es el siguiente:

```

DIJKSTRA (Grafo  $G$ , nodo_fuente  $s$ )
    // inicializamos todos los nodos del grafo. La distancia
    // de cada nodo es infinita y los padres son NULL
    for  $u \in V[G]$  do
        distancia[ $u$ ] = INFINITO
        padre[ $u$ ] = NULL
    next for
        distancia[ $s$ ] = 0
        //encolamos el nodo_fuente  $s$ 
        Encolar (cola, grafo)
    mientras cola no es vacía do
        // Se extrae el nodo que tiene distancia mínima y se
        // conserva la condición de Cola de prioridad
         $u$  = extraer_mínimo(cola)
        for  $v \in \text{adyacencia}[u]$  do
            if distancia[ $v$ ] > distancia[ $u$ ] + peso ( $u$ ,  $v$ ) do
                distancia[ $v$ ] = distancia[ $u$ ] + peso ( $u$ ,  $v$ )
                padre[ $v$ ] =  $u$ 
            fin-if
        Next for
    fin-mientras

```

A continuación, de una manera gráfica, se detalla el algoritmo ya que se considera una manera más didáctica para su comprensión.

ALGORITMO DIJKSTRA	
FASES	DESCRIPCIÓN
	<p>Se quiere encontrar la ruta de peso mínimo entre los puntos A y H, en el grafo de la izquierda.</p>
	<p>Se parte de A buscando los posibles nodos que se pueden visitar.</p> <p>Se crean dos listas que contienen, la primera los nodos ya visitados (Rojo) y una segunda con los nodos que se tiene posibilidad de visitar (Verde).</p> <p>Nodos no Explorados : B(2), G(5), F(11)</p> <p>Nodos Explorados : A</p>

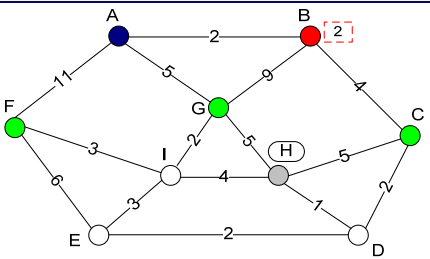
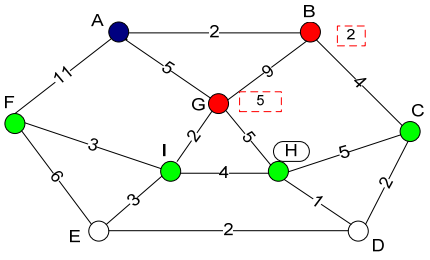
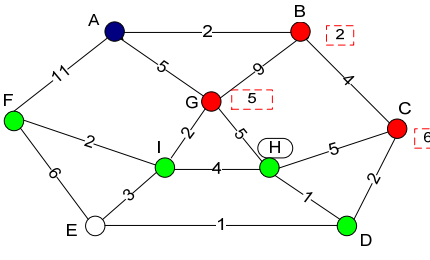
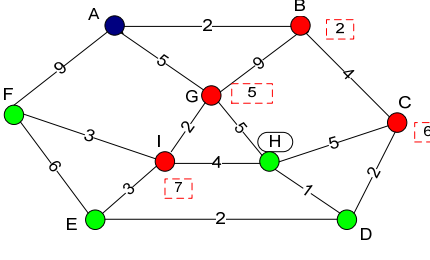
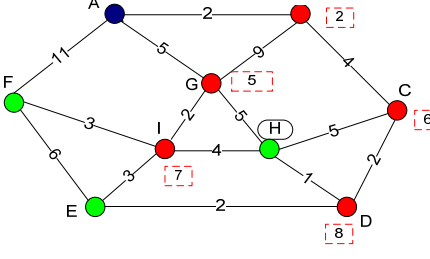
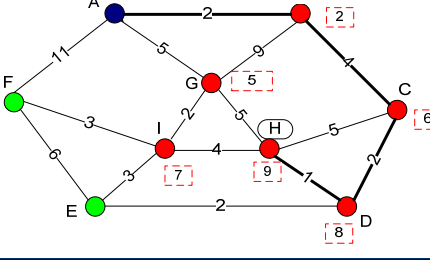
ALGORITMO DIJKSTRA	
FASES	DESCRIPCIÓN
	<p>Se elige de los nodos no visitados el que presente un menor valor al origen. En este caso el B, y se añaden a la lista de nodos no explorados aquellos que sean accesibles desde B.</p> <p>Nodos no Explorados : G(5), F(11), C(6)</p> <p>Nodos Explorados : A, B(2)</p>
	<p>Ahora el nodo no explorado con menor valor es el nodo G, añadiéndose a nodos explorados. También se añaden como nodos con posibilidad de visita los nodos H e I.</p> <p>Nodos no Explorados : F(11), C(6), I(7), H(10)</p> <p>Nodos Explorados : A, B(2), G(5)</p>
	<p>El nodo C presenta un menor valor al origen, por lo tanto es el nodo que le corresponde en esta ocasión su paso a nodos explorados, quedando las listas en el siguiente estado:</p> <p>Nodos no Explorados : F(11), I(7), H(10), D(8)</p> <p>Nodos Explorados : A, B(2), G(5), C(6)</p>
	<p>El siguiente nodo más cercano al origen es el nodo I, realizando la mismas operaciones que en los casos anteriores.</p> <p>Nodos no Explorados : F(11), H(10), D(8), E(10)</p> <p>Nodos Explorados : A, B(2), G(5), C(6), I(7)</p>
	<p>El siguiente nodo en explorar es el nodo D, que modifica la distancia al origen del nodo H, pasando de 10 a 9, quedando las listas en el siguiente estado:</p> <p>Nodos no Explorados : F(11), H(9), E(10)</p> <p>Nodos Explorados : A, B(2), G(5), C(6), I(7), D(8)</p>
	<p>El siguiente nodo que presenta menor distancia es el nodo H, nodo destino por lo que con esto finaliza el algoritmo para determinar la menor distancia a dicho punto. En este caso con un valor de 9.</p> <p>Nodos no Explorados : F(11), E(10)</p> <p>Nodos Explorados : A, B(2), G(5), C(6), I(7), D(8), H(9)</p>

Tabla 1 – Algoritmo de Dijkstra. Ejemplo.

Existen variantes del algoritmo expuesto, entre ellos destacan los siguientes:

- Algoritmo de **Bellman-Ford** que admite la asignación de pesos negativos en los arcos, aunque no de ciclos negativos [VAL00].
- Algoritmo de **Floyd-Warshall**. A veces no es suficiente calcular las distancias con respecto a un vértice s , si no que necesitamos conocer la distancia entre cada par de vértices. Para ello se puede aplicar reiteradamente alguno de los algoritmos anteriores, variando el vértice s de partida. La idea básica del algoritmo es la construcción de una sucesión de matrices W^0, W^1, \dots, W^n , donde el elemento ij de la matriz W^k indique la longitud del camino mínimo $i \rightarrow j$ utilizando como vértices interiores del camino los del conjunto $\{v_1, v_2, \dots, v_k\}$. La matriz W^0 es la matriz de pesos del digrafo, con $w_{ij}^0 = w(ij)$ si existe el arco $i \rightarrow j$, $w_{ii}^0 = 0$ y $w_{ij}^0 = \infty$ si no existe el arco $i \rightarrow j$.

2.2.3 BÚSQUEDA HEURÍSTICA.

Los métodos heurísticos se utilizan para obtener soluciones útiles en problemas sujetos al fenómeno de la explosión combinatoria. Los problemas pueden ser de dos tipos, de tipo P son aquellos que pueden ser resueltos mediante un procedimiento de complejidad polinómica, por otro lado los problemas NP, en los que todos los algoritmos requieren un tiempo exponencial en el peor caso.

La función heurística es una aplicación del espacio de estados sobre un espacio numérico $f(\text{estado}_i) = n_i$, siendo n_i el valor numérico que refleja en qué grado se considera prometedor un estado en la búsqueda de la solución.

2.2.3.1 BÚSQUEDA EN ESCALADA.

Se trata de buscar el camino con máxima pendiente, el parámetro que la define es la *función de evaluación* aplicada, que llamaremos f .

El procedimiento sería el siguiente:

1. Denominar m al estado inicial del problema y asignar m a una variable llamada *elegido*.
2. Hasta que se encuentre una meta o se devuelva fallo realizar las siguientes acciones:
 - 2.1. Expandir m creando el conjunto de todos sus sucesores.

Para cada operador y forma de aplicación:

- a. Aplicar el operador a m generando un nuevo estado.
- b. Si *nuevo* es *meta* salir devolviendo dicho estado.
- c. Si $f(\text{nuevo})$ es mejor que $f(\text{elegido})$, asignar a *elegido* el valor de *nuevo*.

- 2.2. Si $f(\text{elegido}) \neq f(m)$ asignar $m = \text{elegido}$; sino devolver fallo.



- **Ventajas:** La principal es su sencillez. Sólo se precisa almacenar el estado alcanzado (*elegido*) en el proceso de búsqueda. En cada iteración *elegido* será el que tenga un mejor valor de f , no se necesita recordar cuál es el camino recorrido.
- **Desventajas:** La dependencia de la correcta definición de f , exige que la función sea lo más informativa posible, y sencilla de calcular.

2.2.3.2 BÚSQUEDA PRIMERO EL MEJOR.

Consiste en recorrer el grafo de búsqueda eligiendo en cada momento el nodo que tenga un mejor valor para una determinada función heurística, f . Cuando el camino se aleje de la meta se pueden retomar caminos de exploración abandonados previamente. La finalidad es encontrar el camino de menor coste, por lo que la función f debe medir la distancia a la meta.

El procedimiento sería el siguiente:

1. Crear un grafo de exploración G que consista en un único nodo que contiene la descripción del problema. Crear una lista de nodos llamada ABIERTA e inicializarla con dicho nodo.
 2. Crear una lista de nodos llamada CERRADA que inicialmente estará vacía.
 3. Hasta que se encuentre una *meta* o se devuelva *fallo* realizar las siguientes acciones:
 - 3.1 Si ABIERTA está vacía terminar con *fallo*, sino continuar.
 - 3.2 Eliminar el primer nodo de ABIERTA, llamarlo m e introducirlo en CERRADA.
 - 3.3 Expandir m creando punteros en todos sus sucesores.
 - 3.4 Si algún sucesor es *meta* abandonar el proceso iterativo y devolver el camino solución, recorriendo los punteros de sus antepasados.
 - 3.5 Para cada sucesor n de m calcular $f(n)$ teniendo en cuenta las siguientes consideraciones:
 - 3.5.1 Si n es nuevo asociarle el valor $f(n)$ e introducirlo en ABIERTA.
 - 3.5.2 En caso contrario:
 - a. Si $f(n)$ es mayor que el que ya tenía asociado descartar el nuevo nodo.
 - b. Si no, sustituir el valor por el nuevo $f(n)$ y cambiar el puntero a m ; si el nodo estaba en CERRADA, actualizar el coste de sus descendientes ya que podrían ver reducido su valor.
- **Ventajas:** Permite evitar el problema de los mínimos locales que sufre el método de escalada, además asegura que tarde o temprano se encontrara la solución mínima.
 - **Desventajas:** En el procedimiento no se considera el camino recorrido hasta el momento, es decir si un nodo se encuentra cerca de la meta pero se avanza por otro, que a priori tiene una f mejor, en una cierta profundidad se tendrá un menor coste hasta la meta, aunque en realidad el camino total será peor.

2.2.3.3 ALGORITMO A*

Es el mejor método de búsqueda del tipo primero el mejor que sirve para resolver este tipo de problemas, que conlleva el no considerar el camino recorrido hasta un momento dado. La clave del método está en ampliar la definición de la función evaluación para que tenga en cuenta el coste del camino recorrido, resultando la expresión:

$$f(n)=g(n)+h(n)$$

Siendo:

- $g(n)$, coste real del camino de menor coste encontrado hasta el momento desde la raíz del grafo de búsqueda G hasta el nodo en cuestión. Es un estimador del coste real del camino óptimo que se identifica por $g^*(n)$.
- $h(n)$ estimador heurístico del coste del camino óptimo desde el nodo n a una meta. El coste real lo calcula la función $h^*(n)$.
- $f(n)$ es un estimador del coste total del camino óptimo de una solución que pase por el nodo n . El valor real de dicho camino lo devuelve la función inicialmente desconocida, llamada $f^*(n)$.

El procedimiento del algoritmo sería el siguiente [BOT95].

1. Crear una lista de nodos llamada ABIERTA e inicializarla con un único nodo raíz que contiene el estado inicial del problema planteado. Llamar a este elemento r y asignarle $g(r)=0$.
2. Crear una lista de nodos, CERRADA, que inicialmente estará vacía.
3. Repetir hasta que se encuentre la meta o se devuelva *fallo*:
 - 3.1 Si ABIERTA está vacía terminar con fallo, en caso contrario continuar.
 - 3.2 Eliminar el nodo de ABIERTA que contenga un valor mínimo de f , llamar a este nodo m e introducirlo en CERRADA.
 - 3.3 Si m es meta abandonar el proceso iterativo señalado en 3, devolviendo el camino solución, que se obtiene recorriendo los punteros de sus antepasados (creados en 3.5).
 - 3.4 En caso contrario, expandir m generando todos sus sucesores.
 - 3.5 Para cada sucesor n' de m :
 - a. Crear un puntero de n' a m .
 - b. Calcular $g(n') = g(m) + c(m, n')$ | $c(a, b)$: coste de pasar de a a b .
 - c. Si n' está en ABIERTA llamar n al nodo encontrado en dicha lista, añadirlo a los sucesores de m y realizar (c.1).

(c.1) Si $g(n') < g(n)$ entonces redireccionar el puntero de n a m y cambiar el camino de menor coste encontrado a n desde la raíz; $g(n)=g(n')$ y $f(n)=g(n')+h(n)$.

d. Si n' no cumple (c), comprobar si está en CERRADA, llamar n al nodo encontrado en dicha lista y realizar las siguientes acciones:

Si (c.1) no se cumple, abandonar (d), en caso contrario propagar el nuevo menor coste $g(n')$ (por lo que se actualizarán sus valores de f correspondientes) a sus descendientes (que llamaremos n_i | $i = 1, 2, \dots$, siendo sus costes anteriores $g(n_i)$), realizando un recorrido en profundidad de éstos, empezando por n' y teniendo en cuenta las siguientes consideraciones:

1. Para los nodos descendientes n_i cuyo puntero (que debe de apuntar siempre al mejor antecesor hasta ese momento) conduzca hasta el nodo n' , actualizar $g(n_i)=g(n')$ y $f(n_i)=g(n')+h(n_i)$ y seguir el recorrido hasta que se encuentre un n_i que no tenga más sucesores calculados o se llegue a un nodo en que ya ocurra que $g(n_i)=g(n')$; en cuyo caso se habría producido un ciclo y también habría que terminar la propagación.

2. Para los nodos descendientes n_i cuyo puntero no conduzca hacia el nodo n' comprobar si $g(n_i) < g(n')$, en cuyo caso se debe actualizar el puntero para que conduzca hacia el nodo n' (mejor camino desde la raíz encontrado por el momento) y se continúa el proceso de propagación.

e. Si n' no estaba en ABIERTA o en CERRADA, calcular $h(n')$ y $f(n')=g(n')+h(n')$, introducirlo en ABIERTA y añadirlo a la lista de sucesores de m .

En ocasiones se han criticado los métodos de búsqueda heurística por su impredecibilidad. Son capaces de aportar soluciones buenas la gran mayoría de las veces, pero no garantizan encontrar el óptimo. Sin embargo, el método A* posee una serie de propiedades que pueden confirmarse de antemano. Vamos a comentar las más relevantes:

- Es un *método completo* de búsqueda incluso para grafos infinitos. Es decir, termina encontrando la solución cuando ésta exista para cualquier tipo de grafos.
- Es *admisibile*. Es decir, no sólo encuentra una solución sino que la que encuentra es la óptima.

Una función heurística se dice que es admisible si:

$$\forall n \quad h(n) \leq h^*(n)$$

Es decir, si la función heurística que estima la distancia a la meta nunca supera la distancia real existente, entonces el algoritmo A* garantiza encontrar la solución óptima.

- Si la función heurística h es *monótona* y *consistente* el algoritmo encuentra un camino óptimo para todos los nodos expandidos.

Una función es monótona si se cumple:

$$h(n) \leq k(n, n') + h(n') \quad \forall (n, n'),$$

siendo $k(n, n')$ el coste del camino óptimo entre n y n' .

- Entre todos los algoritmos que utilizan una función heurística consistente $h(n)$ y que encuentran una solución óptima, A^* expande el menor número de nodos.
- **Ventajas:** A partir de las propiedades anteriores, se puede decir que el algoritmo A^* es un método completo, admisible y que encuentra la solución óptima para funciones de evaluación heurística consistentes con respecto a la longitud de la solución.
- **Desventajas:** La propiedad de admisibilidad hace que el algoritmo gaste mucho más esfuerzo en discriminar entre caminos prácticamente iguales, por lo que es más realista buscar soluciones dentro de unos márgenes acotados de error. Un inconveniente es el espacio requerido.

2.2.3.4 ESTRATEGIA MINIMAX

Se utiliza en la búsqueda con adversarios. Es un esquema genérico de búsqueda para situaciones en la que el objetivo es ganar una partida en la que participan dos adversarios que realizan movimientos alternativos en el juego. La efectividad de esta estrategia reside en la función heurística aplicada, encargada de medir la ventaja relativa de realizar las distintas jugadas.

El método consiste en prever el máximo número de jugadas posibles y actuar en consecuencia. Para ello recorre un árbol Y/O (en los juegos los enlaces O se utilizan para representar los distintos movimientos realizables en un instante dado; alternativamente, los enlaces Y son utilizados para señalar los movimientos del adversario) llevando a cabo su elección pensando que el adversario va a realizar la mejor de las opciones posibles cuando a él le toque mover. Los nodos del árbol son de dos tipos MAX y MIN, cada uno de éstos toma las decisiones de uno de los contrincantes. El jugador MAX intenta maximizar el valor de la función evaluación (f), por otro lado, los nodos MIN reflejan el modo de actuar que intenta minimizar la función de evaluación.

Para poder etiquetar los nodos con un valor f , se recorre el árbol hasta los nodos hojas (siempre y cuando el árbol sea tratable) o hasta un determinado límite de profundidad. Llegados a este límite, se aplica una función heurística (f) al estado de la partida representado por el nodo (si es un nodo hoja se asigna: ganador=1, perdedor=-1 y empate=0). Los valores de f obtenidos se van subiendo por el árbol aplicando a cada nodo padre el valor máximo o mínimo del valor de sus hijos (dependiendo si es un nodo a jugar por MAX o por MIN), hasta llegar al nodo actual.

El método MINIMAX se comporta como un procedimiento de búsqueda con retroceso con una frontera de exploración calculada. El nivel de exploración depende de

diversos factores entre los que se pueden destacar las características del camino recorrido hasta ese momento y el coste computacional de alcanzar dicha profundidad.

El procedimiento empleado es el siguiente:

MINIMAX (m , profundidad, jugador)

1. Si m no tiene sucesores o se considera alcanzado el límite de profundidad, devolver $mmv(m)=f(m)$.

2. Generar los sucesores de m .

2.1 Inicializar la variable mejor con el valor mínimo que esta pueda tener (puede ser un valor de referencia previamente establecido)

2.2 Para cada sucesor n de m :

2.2.1 $mmv(n)=\text{MINIMAX}(n, \text{profundidad}+1, C(\text{jugador}))$; siendo $C(\text{jugador})$ una función que cambia de jugador.

2.2.2 $\text{mejor} = \max[mmv(n), \text{mejor}]$

3. Una vez analizados recursivamente todos los sucesores de un determinado nivel (indicado por la variable profundidad), se devuelve el que ha obtenido un mejor valor, $mmv = \text{mejor}$

- **Ventajas:** Se trata de un método general y sencillo.
- **Desventajas:** Es poco eficiente al ser demasiado exhaustivo.

2.2.3.5 MINIMAX CON PODA. METODO DE PODA $\alpha-\beta$

Se trata de una modificación del método MINIMAX que permite ahorrarse el recorrido de caminos inútiles del árbol de búsqueda. Para ello recorre el árbol en profundidad (hasta un valor prefijado) y transmitiendo a los niveles superiores del árbol la información obtenida hasta evaluar una hoja, con el propósito de poder abandonar la exploración de algunas ramas.

El método de poda $\alpha-\beta$ se encarga de llevar una anotación para saber cuándo se puede realizar una “poda” en el árbol de búsqueda. Lo que hace es pasar en cada llamada recursiva a un nodo hijo, dos valores (α y β) de manera que α marca la cota inferior de los valores que se van a ir buscando en la parte del árbol que queda por explorar, siendo β la cota superior de esos mismos valores. Si hay algún momento en que el valor de α es mayor o igual al de β , no tendrá sentido seguir con la búsqueda realizándose una poda α si estamos en un nodo MAX o β si el nodo es MIN.

El procedimiento que utiliza es el siguiente:

alfabeta (m , profundidad, jugador, límite_actual, límite_siguiente)

1. Si m no tiene sucesores o si se considera que m ha alcanzado el límite de profundidad, devolver $\alpha\beta v(m)=f(m)$.

2. Generar los sucesores de m .

2.1 Para cada sucesor n de m :

2.2.1 $\alpha\beta v(n) = \text{alfabeta}(n, \text{profundidad}+1, C(\text{jugador}), \text{limite_siguiente}, \text{limite_actual})$,

siendo $C(\text{jugador})$ una función que cambia de jugador.

2.2.2 $\text{limite_siguiente} = \max[-\alpha\beta v(n), \text{limite_siguiente}]$

2.2.3 Si $\text{limite_siguiente} \geq \text{limite_actual}$ entonces abandonar este nivel y devolver limite_siguiente

3. Una vez analizados recursivamente todos los sucesores de un determinado nivel, se devuelve el que ha obtenido un mejor valor.

$$\alpha\beta v(m) = \text{limite_siguiente}$$

La primera llamada al procedimiento se realiza tomando como α el máximo valor que puede tener $f(n)$ y como valor β el valor mínimo de la misma.

- **Ventajas:** Método eficiente que reduce considerablemente el número de nodos expandido y permite afrontar problemas supuestamente intratables.
- **Desventajas:** Dependencia excesiva del ordenamiento de los nodos del árbol que influye en las expansiones de los nodos.

2.3 SOLUCIONES AL PROBLEMA BASADAS EN METAHEURÍSTICAS.

La utilización de algoritmos heurísticos tradicionales actualmente está perdiendo terreno frente a los denominados metaheurísticos. El término heurístico se relaciona con la resolución de problemas de manera inteligente utilizando la información disponible. El término proviene de la palabra griega '*heuriskein*' que significa encontrar o descubrir.

En los métodos heurísticos es tan importante la rapidez para ofrecer una solución, como su calidad, siendo capaces de aportar una buena solución para un tipo de problema bien definido, aprovechando el conocimiento existente.

En las últimas décadas han aparecido técnicas de propósito general con el objetivo de mejorar los resultados de los métodos heurísticos tradicionales, son las denominadas metaheurísticas. Estas se sitúan por encima de las heurísticas, ya que guían el diseño de estas, llegando a utilizar una combinación de ellas para mejorar los resultados. Un posible agrupamiento de las metaheurísticas, en función de las heurísticas utilizadas, puede ser el siguiente:

- **Metaheurísticas constructivas:** Va incorporando elementos a una estructura inicialmente vacía que representa la solución.
- **Metaheurísticas evolutivas:** Son métodos que van construyendo un conjunto de soluciones. El procedimiento consiste en generar, seleccionar, combinar y reemplazar un conjunto de soluciones.

- **Metaheurísticas de búsqueda:** Son métodos que presuponen la existencia de una solución, realizando procedimientos de búsqueda no necesariamente en busca del óptimo.

Existen multitud de técnicas metaheurísticas y algunas se han utilizado para la búsqueda de soluciones en problemas de explosión combinatoria del que forma parte el problema objeto de estudio de este trabajo. Entre las técnicas metaheurísticas que mayor éxito han obtenido en la comunidad científica destacan:

- **Búsqueda Tabú [DRE03]:** Técnica metaheurística que se basa en una búsqueda local o por vecindades, moviéndose de manera iterativa a soluciones vecinas hasta conseguir satisfacer un criterio de parada. Mejora el método heurístico de búsqueda local, evitando que el proceso se detenga en un óptimo local. Esto lo consigue manteniendo una lista de movimientos tabú, evitando acceder a zonas visitadas recientemente.
- **Algoritmos Genéticos:** Introducidos por John Holland en 1970, son algoritmos que se inspiran en la evolución biológica y selección natural [HOL75]. Estos algoritmos hacen evolucionar una población de individuos a semejanza de la evolución biológica natural. Sirven para la resolución de problemas de optimización que pueden ser codificados mediante cadenas finitas en un alfabeto finito. Introducen una función de valoración de los diferentes individuos, denominada “*fitness*”, basada en la función objetivo del problema. Este tipo de algoritmos ha dado muy buenos resultados en diferentes campos con problemas considerados difíciles.
- **Templado Simulado (Simulated Annealing) [DRE03]:** Busca el óptimo global en un espacio de búsqueda grande basándose en el proceso de calentamiento de metales. El proceso consiste en calentar un metal enfriándolo posteriormente de forma controlada, consiguiendo que disminuyan las impurezas. El enfriamiento lento origina una mayor probabilidad de que los átomos se sitúen en estructuras o patrones de mínima entropía. Llevando esto al proceso computacional, se basa en pasar de un estado a otro próximo de manera probabilística en cada iteración, tendiendo a estados de menor energía y avanzando hasta conseguir estados suficientemente aceptables.
- **GRASP [RES06]:** Iniciales de Greedy Randomized Adaptive Search Procedure, metaheurística que presenta dos fases, una primera de construcción que origina una solución buena, aunque no óptima, y otra, a partir de esta, con una búsqueda local donde se examinan vecindades para llegar a un óptimo local en la misma iteración. Este proceso continua hasta alcanzar un criterio de terminación.
- **Optimización basada en Colonia de Hormigas [DOR961][DOR04]:** Algoritmos de inspiración biológica que se basan en el comportamiento de las colonias de hormigas naturales para la resolución de problemas combinatoriales. Es la técnica metaheurística sobre la que se basa el

estudio a realizar y que más detalladamente se describe en el siguiente punto.

Estas técnicas metaheurísticas se basan en algoritmos aproximativos con un propósito general, que combinan de forma inteligente conceptos que permiten explorar y explotar el espacio de búsqueda. Además, este tipo de algoritmos tienen que ser capaces de identificar rápidamente zonas con posibles soluciones de calidad, y no consumir mucho tiempo en aquellas no prometedoras. Esto se consigue equilibrando la exploración de nuevas soluciones y la explotación del conocimiento adquirido.

2.3.1 METAHEURÍSTICA BASADA EN COLONIAS DE HORMIGAS.

En la naturaleza existen numerosas especies que presentan comportamientos autoorganizativos, originando lo que se ha denominado '*inteligencia de enjambres*' (swarm intelligence) [ENG05][KEN01][BON99], campo de investigación que estudia algoritmos inspirados en el comportamiento de colonias de individuos. Como parte de este grupo se encuentran los algoritmos de optimización basados en colonia de hormigas (Ant Colony Optimization, ACO), objeto de estudio del presente trabajo.

Los algoritmos de inspiración biológica basados en colonias de hormigas toman como referencia el comportamiento de este tipo de insectos para la resolución de problemas complejos. Como elementos muy simples, se comunican formando una sociedad capaz de aportar unas soluciones que no son posibles de conseguir individualmente.

Los problemas de rutas y su optimización se adaptan perfectamente a la utilización de algoritmos basados en colonia de hormigas. Este conjunto de problemas, aun los más básicos, son tremendamente difíciles de resolver óptimamente, motivo por lo que la comunidad científica encuentra de gran interés el estudio de este tipo de problemas.

Los problemas resolubles mediante este tipo de algoritmos, tienen en común la determinación del camino mínimo y se pueden caracterizar por los siguientes aspectos:

- El problema presenta un conjunto de restricciones según el modelo elegido.
- Existe un conjunto finito de elementos.
- El problema presenta diversos estados o posibles soluciones.
- Presentan una estructura de vecindario. Existen conexiones entre los diversos elementos.
- Una solución tiene que verificar las restricciones.
- Cada solución presenta un coste.



Todas estas características se presentan en problemas de resolución combinatoria y una posible representación es en forma de grafo. Por tanto, tenemos que:

- El conjunto de elementos finitos son los nodos del grafo.
- Los estados son las posibles soluciones.
- Los arcos representan la estructura de vecindario.
- Cada arco presenta un coste, con lo cual toda solución presentará como coste global la suma de los costes parciales.

2.3.2 COLONIAS DE HORMIGAS NATURALES.

Las hormigas son parte del conjunto de insectos que presentan el fenómeno de autoorganización [BON99], siendo capaces de solventar problemas como adaptación al medio. Constituyen la familia Formiceae, perteneciente al orden Hymenoptera, que destaca por su socialización en la mayoría de las especies que la componen.

Individualmente las hormigas no son capaces de solventar el problema de encontrar la menor distancia entre su hormiguero y la fuente de alimentación. Tampoco existe ningún individuo capaz de coordinar al resto para la resolución del problema, pero entonces, ¿cómo consiguen resolver el problema?. Este comportamiento, en apariencia inteligente, surge del conjunto de individuos, no mostrándose de forma individual. Prevalece el desarrollo colectivo sobre el individual.

Actúan como iteraciones entre elementos de bajo nivel que producen la resolución de complejos problemas a nivel global. Esta iteración se produce a través del medio estableciendo un flujo de información entre los miembros de la colonia. Este es el modelo de comunicación de las hormigas [GOS89], una comunicación por el medio ambiente, denominado "*estigmergia*".

Este comportamiento social no sólo se da en las hormigas, existen otras muchas especies que lo utilizan para solventar sus necesidades y de esta manera, mediante la cooperación de individuos, lograr sus metas. Todas las especies que poseen este comportamiento presentan tres condiciones:

- Un objetivo
- Reglas de interacción
- Un universo donde interactuar.

En el caso de las hormigas su objetivo es la alimentación, la búsqueda del camino más corto entre el hormiguero y la fuente de alimentación. En cuanto a las reglas de interacción, son las que rigen el comportamiento para la toma de decisiones con el fin de conseguir el objetivo, y como tercer punto se encuentra el universo donde interactuar, en este caso el medio por donde se desplazan y comunican.

Las hormigas utilizan como forma de comunicación el depósito en el medio de sustancias químicas denominadas feromonas, siendo estas detectadas por el conjunto de

miembros de la misma especie [GOS89]. Las hormigas tienden a seguir caminos con altos contenidos en feromonas, siendo esta la base de su desplazamiento y actuando siempre de una manera local, es decir, sólo detectan la feromona cuando se encuentran en contacto con ella.

2.3.2.1 DEPÓSITO DE FEROMONAS

Las feromonas son utilizadas por infinidad de especies para su comunicación cumpliendo un papel importantísimo, ya que representa un sistema de comunicación química entre individuos de la misma especie sobre el estado fisiológico, reproductivo o social, las cuales son recibidas e interpretadas por el conjunto de la especie. Estas pueden ser liberadas de forma activa por la especie para indicar ciertas conductas específicas en momentos determinados (emisión de feromona ocasionado por el daño producido en una colonia de hormigas por un ataque externo), o por el contrario de una forma pasiva cuando se dan ciertas circunstancias y el individuo emisor no participa de manera activa (marca de feromona realizada por la hormiga por el simple hecho de desplazarse).

La hormiga, por lo tanto, en su transito es capaz de ir realizando el depósito de feromona en el medio, efectuando por lo tanto, una marca del recorrido elegido. Este hecho es el que origina el aprovechamiento de la información por el resto de la comunidad.

La detección de la feromona depositada por el conjunto de individuos de un hormiguero es la fuente de información que utiliza un individuo para su movimiento. Esto hace que dependiendo de la intensidad de feromona depositada la hormiga decida el camino a recorrer, a mayor cantidad más probabilidad tiene de elegir ese rastro. Este comportamiento queda demostrado en experimentos realizados por biólogos, donde las hormigas prefieren de manera probabilística los caminos con un valor de feromona más elevado.

El depósito continuado origina un sistema de refuerzo del recorrido al añadir el individuo una nueva cantidad de feromona sobre ruta elegida. Este mecanismo hace posible que las hormigas encuentren el camino más corto entre el hormiguero y su fuente de alimento.

Simon Goss en 1989 [GOS89], realizó un experimento para comprobar el comportamiento real de las hormigas, dispuso dos rutas para llegar al destino, una más larga que la otra, y comprobó como evolucionaba en el tiempo el comportamiento de las hormigas.

En un primer momento, cuando la hormiga llega a un punto todavía no visitado, esta elige el camino a seguir de una manera completamente aleatoria, no tiene ningún tipo de información para elegir el camino, y por lo tanto, las opciones de elección son las mismas para cada posible ruta a seguir.

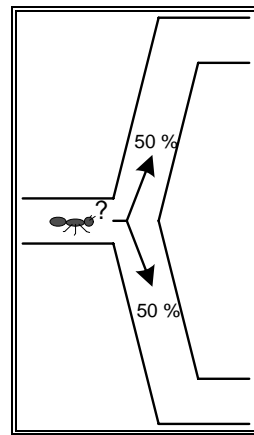
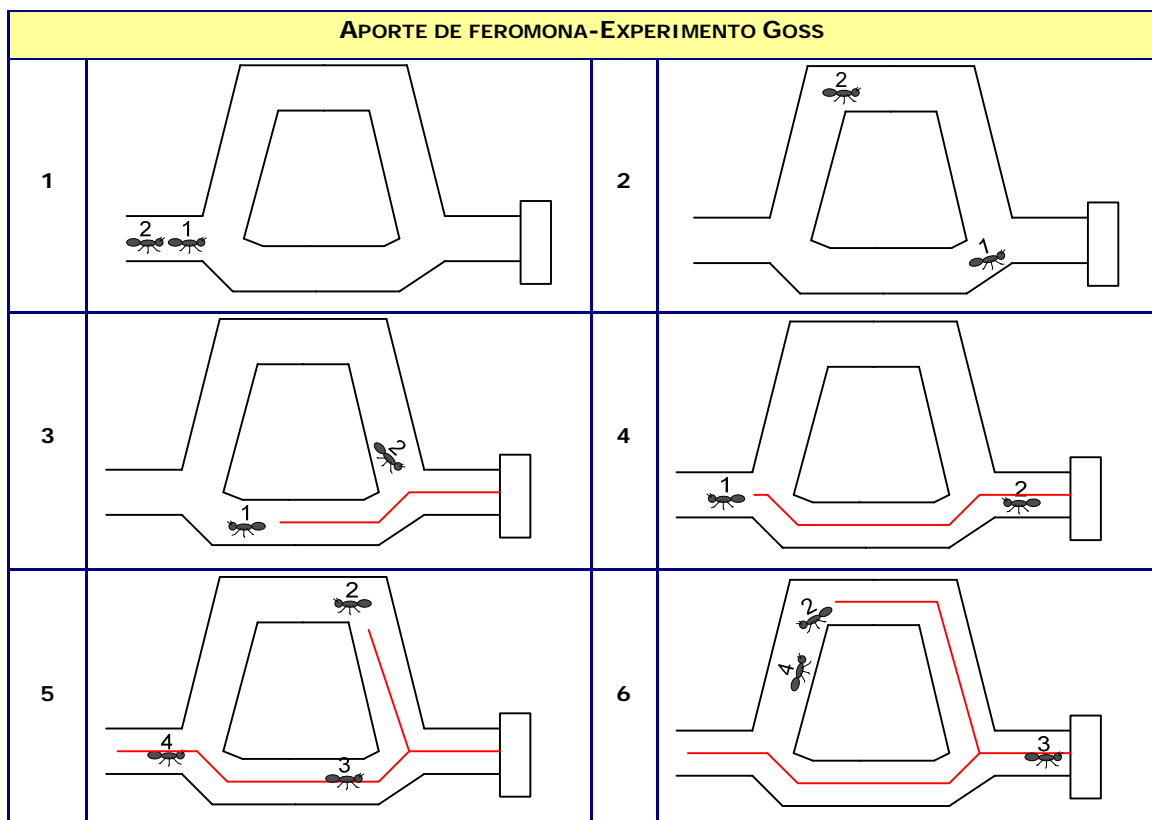
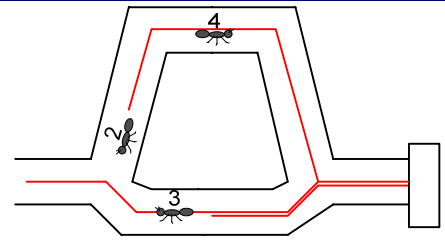
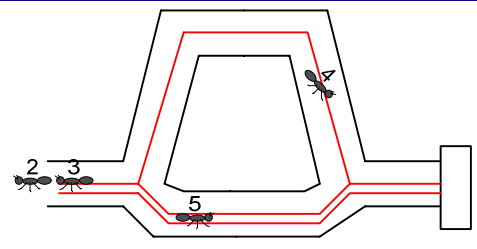


Ilustración 9 – Comportamiento de la hormiga. Decisión de ruta -Inicial-.

En este primer momento según llegan las hormigas, estas tienen las mismas opciones de repartirse entre ambos caminos, ya que al principio la cantidad depositada de feromona es nula, dejando la posibilidad de exploración al conjunto de hormigas para la búsqueda de cualquier posible camino.

En el siguiente ejemplo gráfico se puede observar como las distancias más cortas a la fuente de alimentación originan un aporte de feromona mayor a la ruta.



APORTE DE FEROMONA-EXPERIMENTO GOSS	
7	
8	

Como se puede observar, en la evolución de la deposición de feromona en el tiempo mostrado en el ejemplo, el camino más corto ya presenta el doble de feromona. Esto origina que la probabilidad de un individuo de la colonia de elegir el camino más corto sea cada vez mayor.

Para mejor comprensión, en este ejemplo el depósito de feromona sólo se efectúa en el regreso de la hormiga desde la fuente de alimentación.

Tabla 2 – Aporte feromona. Experimento de Simon Goss.

Una vez detectada la comida, las hormigas se desplazan de nuevo al hormiguero. Las hormigas con un recorrido más corto lo realizarán en un tiempo menor, dando la posibilidad de reforzar con feromona el camino más rápidamente que otras hormigas que han elegido peor ruta. Esto origina una retroalimentación o refuerzo positivo causante de encontrar la ruta más corta entre la fuente del alimento y el hormiguero.

Transcurrido un tiempo las hormigas tienden a confluir por la ruta más corta como se observa en la siguiente ilustración. También indicar que siempre existe la posibilidad de exploración sobre rutas con menor aporte de feromona, ya que la elección de la hormiga es aleatoria en base a las proporciones de feromona detectadas en cada ruta como se verá más adelante.

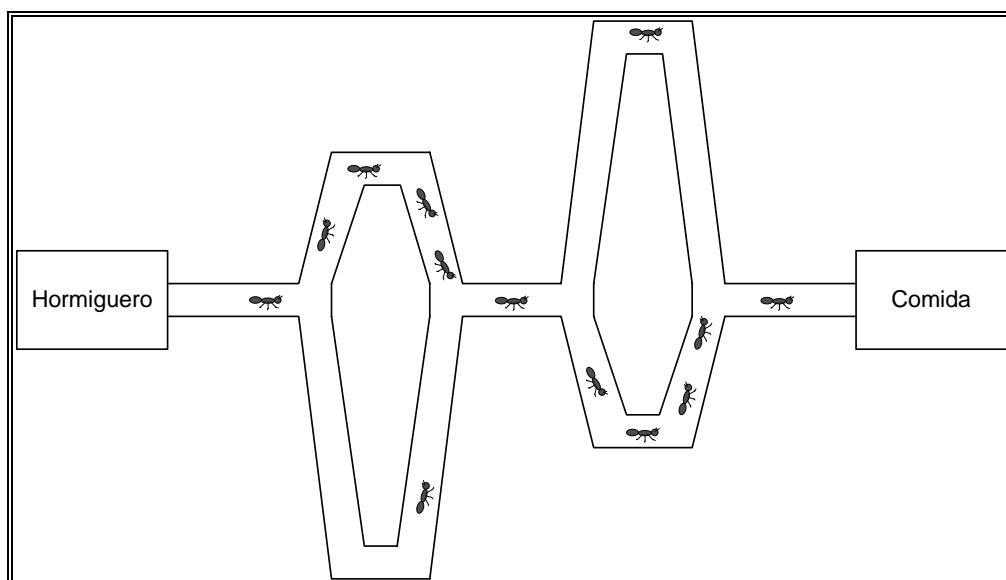


Ilustración 10 – Comportamiento de la hormiga. Decisión de ruta -final-.

Simon Goss a partir de los resultados obtenidos, desarrollo un modelo para reflejar el comportamiento observado. Si una hormiga se sitúa en una bifurcación después de t unidades de tiempo desde el inicio del experimento, y sabiendo que m_1

hormigas han utilizado la ruta 1 y m_2 la ruta 2, la probabilidad de que la hormiga elija una de las dos rutas es la siguiente [GOS89]:

$$p_{1(m+1)} = \frac{(m_1 + k)^h}{(m_1 + k)^h + (m_2 + k)^h}$$

$$p_{2(m+1)} = 1 - p_{1(m+1)}$$

Donde k y h son parámetros del modelo que sirven de ajuste a los datos del experimento.

Como complemento, también hay que indicar que la feromona en un entorno natural, sufre una evaporación con el paso del tiempo, así los recorridos menos valiosos presentan una pérdida progresiva de feromona que les hace cada vez menos interesantes para la colonia.

Si este modelo lo queremos trasladar al entorno computacional, hay que crear unos elementos capaces de simular el comportamiento de las hormigas naturales. Deben tener un objetivo definido, unas reglas de comportamiento y un modo de comunicación. Deben ser elementos independientes, capaces de interactuar con el medio para establecer una comunicación entre ellos.

Además del comportamiento, hay que simular el medio donde habitan y como construyen una solución, para ello simulamos el entorno donde la hormiga se va desplazando hasta conseguir la solución.

2.3.2.2 SISTEMA CONTADOR DE PASOS.



Ilustración 11 – Cataglyphis fortis. Zancos de modificación de amplitud de paso.

Otro comportamiento natural de las hormigas que se utilizará en la elaboración del proyecto y que se ha considerado necesario hacer una breve mención en este punto, son los estudios realizados sobre la especie *Cataglyphis fortis* [APA07][MOR00], especie que habita en el desierto del Sahara y en la que se ha descubierto la presencia de una especie de podómetro que les indica la distancia en pasos que hay entre la fuente de alimentación y el hormiguero.

Hay que indicar, que por causa de la climatología existente, el depósito de feromonas no es una buena fuente de comunicación debido a la evaporación, muy elevada por las altas temperaturas del hábitat donde se desenvuelve. Esta carencia, en su



evolución, la ha sabido sustituir mediante otra técnica de supervivencia, el contador de pasos. Esta hormiga es capaz de regresar en línea recta al hormiguero a pesar de haber recorrido un camino sinuoso hasta la fuente de alimentación, es decir en todo momento sabe la distancia exacta que le separa de su hormiguero.

Se sabía que son capaces de conocer la dirección correcta por el patrón de luz polarizada del cielo, pero los mirmecólogos se preguntaban como sabían la distancia a recorrer. Un trabajo publicado por la revista Science [MOR00] sobre un trabajo de Harald Wolf, demuestra que esta especie presenta una simulación de podómetro o contador de pasos. Teniendo como base unas claves visuales del cielo y la distancia precisa que le separa del hormiguero, pueden regresar en línea recta y utilizarlo para posteriores expediciones.

El experimento consistía en observar el comportamiento de las hormigas en una distancia fija, en este caso diez metros, cuando se les alargaban las patas o se les cortaban, es decir se les modificaba la amplitud de un paso. El experimento fue un éxito, en la vuelta, a las hormigas que no se les realizaba ninguna manipulación recorrían la distancia exacta, mientras que a las que se le alargaron las patas presentaron un recorrido mayor, exactamente 15 metros.

Esto venia a demostrar que al parecer las hormigas, aunque son especies muy simples, presentan cierta capacidad para memorizar los pasos que realizan y con una determinada orientación ser capaces de llegar al hormiguero.

Estos comportamientos inspirarán una vía de trabajo sobre depósitos de feromona en base a una referencia única que posteriormente se detallará.

2.3.3 COLONIAS DE HORMIGAS ARTIFICIALES.

Una vez expuestas las bases sobre los comportamientos biológicos que inspiran el modelo, presentamos el primer algoritmo de este tipo que fue denominado "Ant System" [DOR96] para la resolución del problema de vendedor viajante, ya que es uno de los problemas de optimización combinatoria más conocido, obteniendo unos resultados no muy afortunados, pero que causaron la curiosidad de gran número de investigadores que fueron modificando el algoritmo y aplicando esta técnica a este conjunto de problemas.

El algoritmo consistía en la simulación del movimiento y comunicación a través del medio de una colonia de hormigas naturales. Como características comunes se pueden destacar:

- Tanto las hormigas naturales como artificiales son elementos simples, formando una población capaz de trabajar de forma coordinada para alcanzar su objetivo.
- Las hormigas naturales tienen como meta la búsqueda de comida, mientras que las artificiales buscan la solución a un problema de optimización.

- Un elemento en particular puede encontrar una solución, pero es la cooperación de todos los individuos la que origina la mejora de los resultados.

Hay que indicar, como se verá posteriormente de una forma más detallada en la explicación de los algoritmos, que las hormigas artificiales presentan una mayor versatilidad de acciones, todo ello debido a la experiencia en experimentos realizados y con el fin de mejorar los resultados aportados por los algoritmos. Por esta circunstancia, existen diferentes modelos a la hora de depositar la feromona e incluso la utilización de mecanismos de otros campos de la inteligencia artificial, todo ello persiguiendo la optimización de los algoritmos desarrollados.

Hay que construir un modelo que represente el ambiente natural de las hormigas y que sea capaz de soportar su comportamiento. Para la representación del medio se utilizan los grafos, siendo estos capaces de plasmar todos los requisitos necesarios. La construcción del modelo toma como base lo siguiente:

- Las hormigas se mueven de un nodo a otro a través de los arcos.
- Las hormigas depositan la feromona en los arcos por donde pasan.
- La cantidad de feromona depende de la solución encontrada.
- Las hormigas eligen su desplazamiento de forma aleatoria en función de la feromona detectada en el arco correspondiente y la distancia al nodo destino.
- Sólo se puede visitar un determinado nodo una sola vez.

Cada hormiga artificial es un elemento probabilístico que realiza la construcción de una solución imitando a la hormiga natural, además tiene que ser capaz de recordar el recorrido realizado, la heurística utilizada y la feromona detectada, para tomar la decisión del camino a seguir.

El medio natural también sufre una evaporación de feromona que el sistema artificial tiene que ser capaz de reflejar, de esta manera se limita el crecimiento de rastros de feromona y que la solución adoptada corresponda a un óptimo local.

2.3.4 MODELOS DE OPTIMIZACIÓN BASADOS EN COLONIAS DE HORMIGAS.

Una vez indicadas las características que tiene que ser capaz de cumplir el sistema, se procede hacer un repaso de la evolución que han sufrido este tipo de algoritmos mediante la explicación de los más representativos basados en colonias de hormigas.

2.3.4.1 *ANT SYSTEM. ALGORITMO BÁSICO.*

Tal y como se ha mencionado, fue el primer algoritmo de optimización de colonia de hormigas, desarrollado por M. Dorigo, V. Maniezzo, A. Colomi en 1991 [DOR96][DOR04]. Este algoritmo se realizó para la resolución del problema del viajante de comercio, problema de optimización combinatoria más tradicional y estudiado, buscando la ruta más corta para viajar a un conjunto de ciudades.



El algoritmo trata de simular el comportamiento de las hormigas naturales. Este debe de ser capaz de admitir las tareas que realiza la hormiga y simular el comportamiento del conjunto.

Se realizaron tres variantes del algoritmo según la forma de realizar el depósito de feromona en el medio.

- **Densidad:** Se realizaba el depósito de feromona durante el transcurso del recorrido (actualización en línea paso a paso de feromona). La cantidad de feromona depositada era siempre constante.
- **Cantidad:** Se realizaba el depósito de feromona durante el transcurso del recorrido (actualización en línea paso a paso de feromona). La cantidad de feromona depositada estaba relacionada a la deseabilidad heurística del tramo.
- **Ciclo:** El depósito de feromona se realiza una vez finalizada una solución (actualización en línea a posteriori de feromona).

Esta última variante es la que mejor resultados proporcionó y la que se conoce por Ant System (AS) ó Sistema de Hormigas (SH).

El SH se caracteriza por el hecho de que la actualización de feromona se realiza una vez que todas las hormigas han completado sus soluciones, y se lleva a cabo como sigue: primero, todos los rastros de feromona se reducen en un factor constante, implementándose de esta manera la evaporación de feromona. A continuación cada hormiga de la colonia deposita una cantidad de feromona que es función de la calidad de su solución. Inicialmente, el SH no usaba ninguna acción en un segundo plano, pero es relativamente fácil, por ejemplo, añadir un procedimiento de búsqueda local para refinar las soluciones generadas por las hormigas.

A grandes rasgos el algoritmo presenta el siguiente desarrollo:

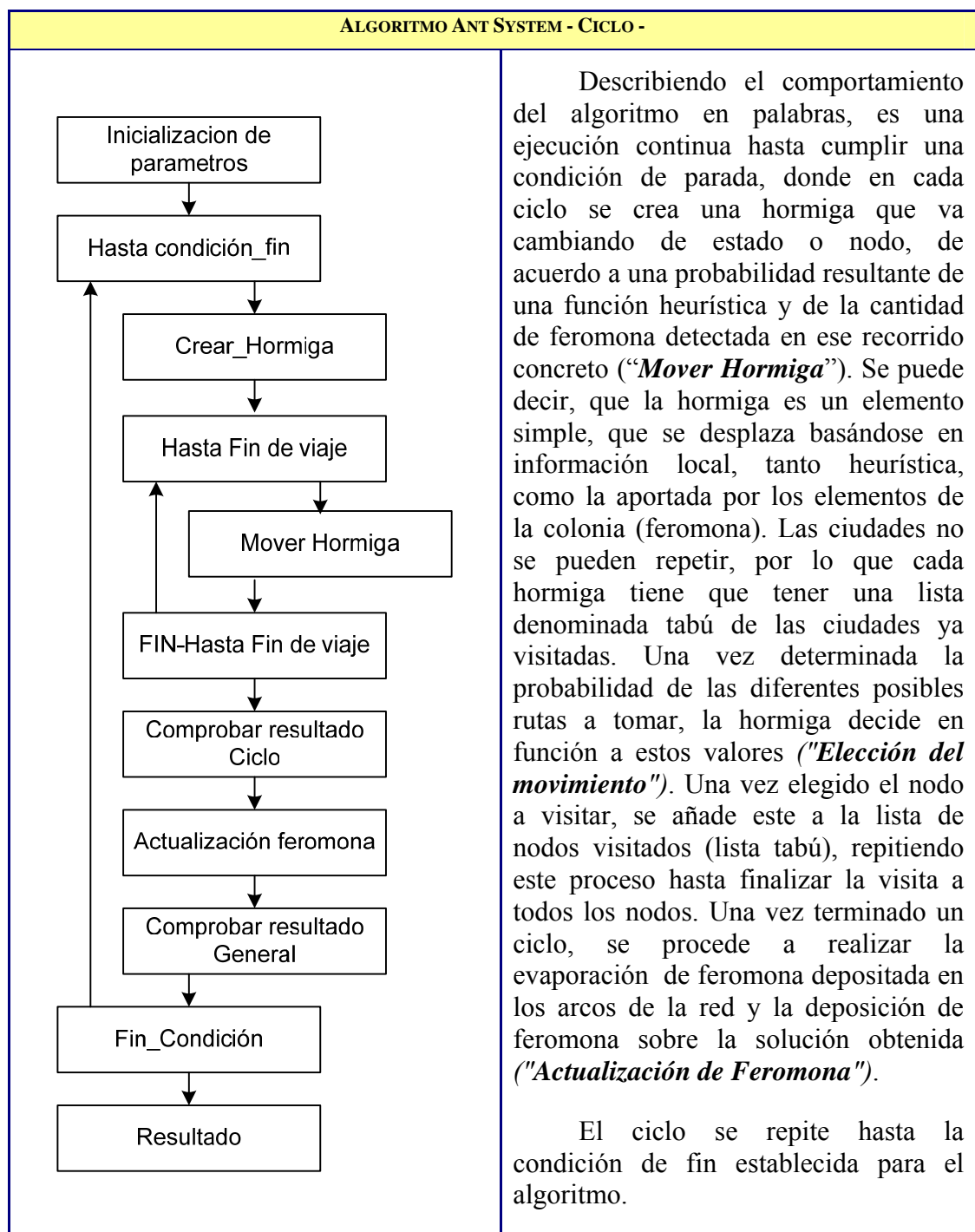


Tabla 3 - Algoritmo Ant System.

Se han realizado multitud de variaciones del algoritmo principalmente en las funciones de acción sobre el tratamiento de la feromona.

Su pseudocódigo podría ser el siguiente:

```
Procedimiento AS ()
  Inicializar parámetros
  Mientras condicion do
    Inicializar_Nuevahormiga()
    Mientras estado <> estado_fin
      Por cada_arco_posible_movimiento
        Calcular probabilidad_eleccion
      Fin_Por
      Siguiente_posicion = politica_decision
      Lista_Posiciones += Siguiente_posicion
    Fin_mientras
  Realizar_evaporacion
  Depósito_feromona(lista_Posiciones)
Fin_mientras
Fin_procedimiento
```

Las hormigas utilizan el depósito de feromonas para recordar su comportamiento, es decir, acumular el conocimiento que van adquiriendo del problema a resolver. En un primer momento, todos los arcos presentan la misma probabilidad y para ello se considera oportuno introducir un pequeño valor de feromona, cantidad que hace posible que caminos sin explorar también tengan probabilidad de ser recorridos.

Como se han identificado en el algoritmo, existen dos procesos fundamentales en su ejecución:

Elección del movimiento.

- Una hormiga k elige ir al siguiente nodo con una probabilidad que viene dada con la formula[DOR961]:

$$p_{rs}^k = \begin{cases} \frac{[r_{rs}]^\alpha \times [\eta_{rs}]^\beta}{\sum_{u \in N_r^k} [r_{ru}]^\alpha \times [\eta_{ru}]^\beta}, & \text{si } s \in N_k(r) \\ 0, & \text{En otro caso} \end{cases}$$

Donde, $N_k(r)$ son los nodos alcanzables por la hormiga k desde el nodo r .

α y β son parámetros que ponderan la importancia de la heurística utilizada y los valores de feromona detectados.

r_{rs} , representa el rastro de feromona entre los puntos r y s

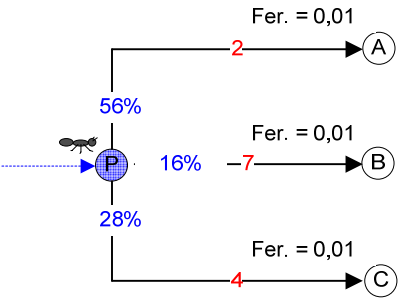
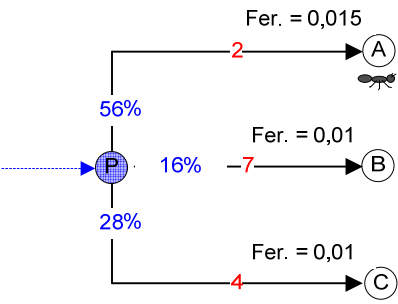
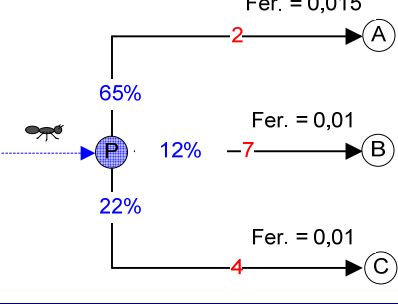
η_{rs} , representa el valor de la función heurística elegida, deseabilidad.

- Cada hormiga k almacena el recorrido realizado para no repetir visitas al mismo nodo.

Como se puede comprobar, el definir unos valores de los parámetros α y β puede hacer variar bastante los resultados. Si hacemos $\alpha = 0$, significa que sólo damos

importancia a la función heurística elegida, mientras que un valor de $\beta = 0$ solamente tiene en cuenta los rastros de feromona detectados por la hormiga. Esto último, provoca que la probabilidad de construir óptimos locales quede muy fortalecida, cosa que no debe de ocurrir para un buen funcionamiento del algoritmo.

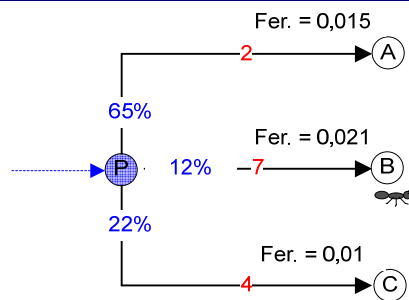
Un ejemplo gráfico simulado puede ayudar a la comprensión de como funciona la elección del destino.

ELECCIÓN DEL MOVIMIENTO	
RUTA REALIZADA	DESCRIPCIÓN
	<p>En un primer momento, cuando la hormiga llega al nodo P, siendo esta la primera en visitarlo, se encuentra con unos valores de feromona iguales para las tres rutas a seguir, en cambio las distancias a los siguientes puntos no son las mismas.</p> <p>Los valores de probabilidad para cada una de las tres posibles rutas para un valor de 1 tanto para α y β, son los siguientes:</p> $P_{P-A} = \frac{(0,01)^\alpha \times (1/2)^\beta}{(0,01)^\alpha \times (1/2)^\beta + (0,01)^\alpha \times (1/7)^\beta + (0,01)^\alpha \times (1/4)^\beta} = 0,56$ $P_{P-B} = \frac{(0,01)^\alpha \times (1/7)^\beta}{(0,01)^\alpha \times (1/2)^\beta + (0,01)^\alpha \times (1/7)^\beta + (0,01)^\alpha \times (1/4)^\beta} = 0,16$ $P_{P-C} = \frac{(0,01)^\alpha \times (1/4)^\beta}{(0,01)^\alpha \times (1/2)^\beta + (0,01)^\alpha \times (1/7)^\beta + (0,01)^\alpha \times (1/4)^\beta} = 0,28$
	<p>La hormiga aleatoriamente, según las probabilidades obtenidas, elige el punto A para su ruta hacia el punto destino. Una vez alcanzado el destino, se realiza la actualización de feromona correspondiente. En cada arco perteneciente al recorrido seguido por la hormiga, se origina una variación del valor de la feromona.</p> <p>Feromona (P-A) : 0,01 \rightarrow 0,015</p>
	<p>Cuando llega otra hormiga al punto P tiene que realizar el cálculo de probabilidades con los nuevos valores existentes.</p> <p>Los cálculos para la obtención de los resultados serían los siguientes:</p>

$$P_{P-A} = \frac{(0,015)^{\alpha} \times (1/2)^{\beta}}{(0,015)^{\alpha} \times (1/2)^{\beta} + (0,01)^{\alpha} \times (1/7)^{\beta} + (0,01)^{\alpha} \times (1/4)^{\beta}} = 0,65$$

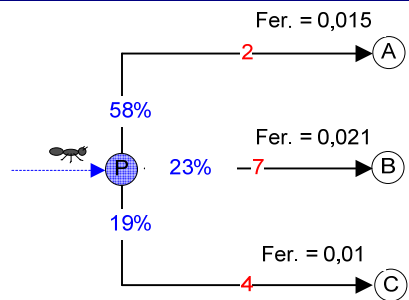
$$P_{P-B} = \frac{(0,01)^{\alpha} \times (1/7)^{\beta}}{(0,015)^{\alpha} \times (1/2)^{\beta} + (0,01)^{\alpha} \times (1/7)^{\beta} + (0,01)^{\alpha} \times (1/4)^{\beta}} = 0,12$$

$$P_{P-C} = \frac{(0,01)^{\alpha} \times (1/4)^{\beta}}{(0,015)^{\alpha} \times (1/2)^{\beta} + (0,01)^{\alpha} \times (1/7)^{\beta} + (0,01)^{\alpha} \times (1/4)^{\beta}} = 0,22$$



La hormiga también puede elegir la que teóricamente es la peor opción, ya que es una elección aleatoria. Este es el caso que se representa, donde la hormiga ha elegido la ruta hacia el punto B. Una vez concluida la ruta hasta el destino, se realiza la actualización correspondiente de feromona en la red.

Feromona (P-B) : 0,01 → 0,021



Una tercera hormiga llega al punto P, en este caso ya se ha actualizado la cantidad de feromona debida a la segunda hormiga. Con estos nuevos valores la hormiga obtiene las siguientes probabilidades de opción de desplazamiento.

$P_{P-A} = \frac{(0,015)^{\alpha} \times (1/2)^{\beta}}{(0,015)^{\alpha} \times (1/2)^{\beta} + (0,021)^{\alpha} \times (1/7)^{\beta} + (0,01)^{\alpha} \times (1/4)^{\beta}} = 0,58$
$P_{P-B} = \frac{(0,021)^{\alpha} \times (1/7)^{\beta}}{(0,015)^{\alpha} \times (1/2)^{\beta} + (0,021)^{\alpha} \times (1/7)^{\beta} + (0,01)^{\alpha} \times (1/4)^{\beta}} = 0,23$
$P_{P-C} = \frac{(0,01)^{\alpha} \times (1/4)^{\beta}}{(0,015)^{\alpha} \times (1/2)^{\beta} + (0,021)^{\alpha} \times (1/7)^{\beta} + (0,01)^{\alpha} \times (1/4)^{\beta}} = 0,19$
<p>Los valores obtenidos, como se ha informado anteriormente, han resultado de los cálculos con unos valores para α y β de 1. Si por el contrario, se quiere dar menos importancia a la función heurística estableceríamos el valor de β por debajo de 1. En el ejemplo siguiente se considera un valor β de 0,7. Con esto se consigue una mayor exploración, las probabilidades de las diferentes opciones tienden a igualarse, dando más importancia a la cantidad detectada de feromona.</p>
$P_{P-A} = \frac{(0,015) \times (1/2)^{0.7}}{(0,015) \times (1/2)^{0.7} + (0,021) \times (1/7)^{0.7} + (0,01) \times (1/4)^{0.7}} = 0,50$
$P_{P-B} = \frac{(0,021) \times (1/7)^{0.7}}{(0,015) \times (1/2)^{0.7} + (0,021) \times (1/7)^{0.7} + (0,01) \times (1/4)^{0.7}} = 0,29$
$P_{P-C} = \frac{(0,01) \times (1/4)^{0.7}}{(0,015) \times (1/2)^{0.7} + (0,021) \times (1/7)^{0.7} + (0,01) \times (1/4)^{0.7}} = 0,21$

Tabla 4 – Elección de movimiento.

Actualización de la feromona

Presenta dos subprocesos:

- *Evaporación.* Los rastros de feromona se reducen un valor constante. Esto es lo que representa la evaporación de la feromona del sistema natural.

Los diferentes arcos sufren una disminución de su valor de feromona que viene dado por la expresión.

$$r_{rs} \leftarrow (1 - \rho) \times r_{rs}$$

, siendo $\rho \in (0,1]$ el factor de evaporación

- *Deposición.* A continuación se realiza el depósito de feromona en la ruta seguida por la hormiga en función de la solución obtenida.

$$r_{rs} \leftarrow r_{rs} + \Delta r_{rs}^k, \quad \forall a_{rs} \in S_k$$

Donde $\Delta r_{rs}^k = f(C(S_k))$, es la cantidad de feromona que se deposita, función de la calidad de la solución obtenida. La calidad de una solución normalmente viene representada por el inverso de la distancia recorrida.

Esta ha sido la base para multitud de modelos desarrollados en un intento de hacer más eficaces este tipo de algoritmos. A continuación se informa de aquellos que, partiendo de este modelo, han obtenido un mayor éxito.

2.3.4.2 ANT SYSTEM Y ELITISMO

Surgió como consecuencia del esfuerzo de Marco Dorigo para mejorar el algoritmo existente [DOR96]. Se basa en el modelo explicado anteriormente, pero añadiendo una característica que conseguía unos mejores resultados. La extensión consistía en un aporte extra de feromona en la solución con mejor resultado obtenido hasta el momento.

Este aporte era función de la solución considerada y de e , que representa el número de hormigas elitista tenidas en cuenta,

$$r_{rs} = r_{rs} + e \times f\left(C\left(S_{\text{mejor-global}}\right)\right)$$

2.3.4.3 ANT COLONY SYSTEM (ACS)

Algoritmo que surge de la extensión de los anteriores algoritmos y la utilización de técnicas de aprendizaje como el Q-learning [GAM95][DOR96], publicado por Dorigo y Gambardella en 1997 [DOR97], donde se realiza la modificación del algoritmo en cuatro facetas principalmente:

- La transición de estados, ofreciendo un balance entre la exploración y explotación del conocimiento acumulado. Se modifica para permitir explícitamente la exploración. La regla utilizada se denomina regla proporcional pseudo-aleatoria.

Sea k una hormiga situada en el nodo r , q_0 ($0 \leq q_0 \leq 1$) un parámetro y q un valor aleatorio en $[0,1]$, el siguiente nodo s se elige aleatoriamente mediante la siguiente distribución de probabilidad.

Si $q \leq q_0$

$$p_{rs}^k = \begin{cases} 1, si & s = \arg \max_{u \in N_k(r)} \{ \tau_{ru} \times \eta_{ru}^\beta \} \\ 0, en & otro \quad caso \end{cases}$$

Si $q > q_0$

$$p_{rs}^k = \begin{cases} \frac{[\tau_{rs}]^\alpha \times [\eta_{rs}]^\beta}{\sum_{u \in N_r^k} [\tau_{ru}]^\alpha \times [\eta_{ru}]^\beta}, & si \quad s \in N_k(r) \\ 0, & en \quad otro \quad caso \end{cases}$$

- La actualización global de feromona, ya que realizan una actualización global sobre la mejor ruta encontrada en cada iteración.

Para llevar a cabo la actualización, el sistema de hormigas sólo considera una hormiga concreta, la que generó la mejor solución global, $S_{\text{mejor-global}}$ (aunque en algunos trabajos iniciales se consideraba también una actualización basada en la mejor hormiga de la iteración, en el sistema de colonia de hormigas casi siempre se aplica la actualización por medio de la mejor global). La actualización de la feromona se hace evaporando primero los rastros de feromona en todas las conexiones utilizadas por la mejor hormiga global (es importante recalcar que, en el sistema de colonia de hormigas, la evaporación de feromona sólo se aplica a las conexiones de la solución, que es también la usada para depositar feromona) tal como sigue:

$$\tau_{rs} \leftarrow (1 - \rho) \times \tau_{rs}, \quad \forall a_{rs} \in S_{\text{mejor-global}}$$

A continuación, el se deposita feromona usando la regla:

$$\tau_{rs} \leftarrow \tau_{rs} + \rho \times f(C(S_{\text{mejor-global}})), \quad \forall a_{rs} \in S_{\text{mejor-global}}$$

- La actualización local de feromona. Las hormigas al ir transitando por los arcos, lo que hacen es disminuir la feromona, favoreciendo por lo tanto la exploración. El objetivo es hacer menos atractivos los arcos visitados, buscando la no convergencia en la misma ruta. Su representación matemática sería la siguiente:

$$\tau_{rs} \leftarrow (1 - \varphi) \times \tau_{rs} + \varphi * \tau_0$$

Donde $\varphi \in (0,1]$ es un segundo parámetro de decremento de feromona. Como puede verse, la regla de actualización local incluye tanto la evaporación de feromona como la deposición de la misma. Ya que la cantidad de feromona depositada es muy pequeña (de hecho, τ_0 es el valor del rastro de feromona inicial y se escogiese de tal manera que, en la práctica, se corresponda con el límite menor de rastro de feromona, esto es, con la elección de las reglas de actualización de feromona del sistema de colonia de hormigas ningún rastro de feromona puede caer por debajo de τ_0), la aplicación de esta regla hace que los rastros de feromona entre las conexiones recorridas por las hormigas disminuyan.

Así, esto lleva a una técnica de exploración adicional del sistema de colonia de hormiga ya que las conexiones atravesadas por un gran número de hormigas son cada vez menos atractivas para el resto de hormigas que las recorren en la iteración actual, lo que ayuda claramente a que no todas las hormigas sigan el mismo camino.

- Uso de una lista candidata. Lista donde se tienen ordenadas las ciudades por preferencia de visita.

El algoritmo además añade una búsqueda local, mediante el método 3-opt propuesto por Bock en 1958, realizando un intercambio de arcos con el fin de reducir la longitud encontrada. Este método, realiza tres cortes en el camino e intercambia las ciudades destino y de esta manera no invertir el sentido de las ciudades visitadas.

El algoritmo diseñado según este modelo es el siguiente:

```
Procedimiento ACS ()  
'Fase Inicialización  
    Inicializar parámetros  
    Inicializar feromona arcos  
'Fase Ejecución  
    Mientras no se cumpla condición fin  
        Mientras no este completa lista_tabú  
            Si es última Posición en lista_tabú  
                Insertar ciudad origen  
            Sino  
                Por cada hormiga  
                    Elegir próxima ciudad  
                    Mover hormiga  
                    Insertar Ciudad en Tabú  
                Fin_por  
            Fin_si  
                Actualizar feromona local  
        Fin_mientras  
        Por cada hormiga  
            Ver longitud ruta  
            Actualizar feromona global  
        Fin_por  
    Fin mientras
```

2.3.4.4 MAX-MIN ANT SYSTEM, (MMAS)

Surge de la observación del comportamiento de algoritmos anteriormente citados [STU00]. Este algoritmo, modifica la estrategia a la hora de realizar el depósito de feromonas y solamente permite la actualización a las hormigas que alcanzan buenos resultados. Fue presentado por Tomas Stützle y Holger Hoos [STU96],

Esta estrategia puede dar origen a una convergencia prematura en un resultado no óptimo, pero se consideraba oportuno que los mejores resultados ejercieran una influencia mayor en el algoritmo utilizado. Para evitar el aumento desmesurado del valor de feromona en rutas no óptimas y por lo tanto la convergencia en estas rutas, se introdujo el concepto de máximo y mínimo, que se referían a los valores que podían alcanzar las rutas en cuanto a cantidad de feromona depositada. El nivel mínimo pone límite al nivel de exploración, para que este siempre tenga posibilidades de ejecución.

$$r_t^{min} \leq r_{rs,t+1} \leq r_t^{max}$$

El algoritmo introducía otro nuevo concepto, era la posibilidad de volver a inicializar los valores de feromona depositada, imponiendo una nueva exploración del espacio de búsqueda.

Extiende el Sistema de Hormigas en los siguientes aspectos

- Se aplica una actualización de los rastros de feromona fuera de línea, de manera similar a como se hace en el Sistema de Colonias de Hormigas. Después de que todas las hormigas hayan construido su solución cada rastro de feromona sufre una evaporación:

$$\tau_{rs} \leftarrow (1 - \rho) \times \tau_{rs}$$

A continuación la feromona se deposita siguiendo la siguiente fórmula:

$$\tau_{rs} \leftarrow \tau_{rs} + f(C(S_{mejor})) \quad \forall a_{rs} \in S_{mejor}$$

La mejor hormiga a la que se le permite añadir feromona puede ser la que tiene una solución mejor de la iteración o la solución mejor global. Los resultados experimentales demuestran que el mejor rendimiento se obtiene incrementando gradualmente la frecuencia de escoger la mejor global para la actualización de feromona.

Además, las soluciones que ofrecen las hormigas suelen ser mejoradas usando optimizadores locales antes de la actualización de feromona.

- Los valores posibles para los rastros de feromona están limitados al rango $[\tau_{min}, \tau_{max}]$. Por lo tanto, la probabilidad de un estancamiento del algoritmo disminuye al darle a cada conexión existente una probabilidad, aunque bastante pequeña, de ser escogida. En la práctica, existen heurísticas para fijar los valores de τ_{min} y τ_{max} . Se puede ver que, a causa de la evaporación de la feromona, el nivel máximo de feromona en los rastros está limitado a $\tau_{max} = 1/(\rho \times C(S^*))$, donde S^* es la solución óptima. Basándonos en este resultado, la mejor solución global puede usarse para estimar τ_{max} sustituyendo S^* por $S_{mejor-global}$ en la ecuación de τ_{max}^* . Para τ_{min} , normalmente sólo es necesario escoger su valor de tal manera que sea un factor constante menor que τ_{max} .

Para poder incrementar la exploración de nuevas soluciones, el MMAS utiliza en ocasiones re-inicializaciones de los rastros de feromona.

- En vez de inicializar los rastros de feromona a una cantidad pequeña, el MMAS los inicializa a una estimación del máximo permitido para un rastro (la estimación puede obtenerse generando una solución S' con una heurística voraz y reemplazando dicha solución S' en la ecuación de r_{max}^*). Esto lleva a una componente adicional de diversificación en el algoritmo,

ya que al comienzo las diferencias relativas entre los rastros de feromona no serán muy acusadas, lo que no ocurre cuando los rastros de feromona se inicializan a un valor muy pequeño.

2.3.4.5 SISTEMA MEJOR-PEOR HORMIGA. (SMPH)

Este sistema fue propuesto por Cordón y otros en 1999, es un algoritmo de sistema de hormigas que incorpora conceptos de computación evolutiva. Este algoritmo es otra extensión del Sistema de Hormigas [ALO00].

El SMPH utiliza la misma regla de transición de estados que el Sistema de Hormigas, así como la misma regla de evaporación de feromona que se aplica a todas las transiciones.

Además, tal como hace el MMAS, el SMPH siempre considera la explotación sistemática de optimizadores locales para mejorar las soluciones de las hormigas. Formando el núcleo del SMPH podemos encontrar las siguientes tres acciones durante la ejecución:

- La regla mejor-peor de actualización de rastros de feromona, basada en la regla de actualización del vector de probabilidades, refuerza las aristas que se encuentran en la mejor solución global. Además, penaliza cada conexión de la peor solución generada hasta el momento, $S_{\text{peor-actual}}$, que no se encuentre en la mejor global realizando una evaporación de feromona adicional de esos rastros. Por tanto, la regla de actualización de feromona en el SMPH se convierte en:

$$\tau_{rs} \leftarrow \tau_{rs} + \rho \times f(C(S_{\text{mejor-global}})), \quad \forall a_{rs} \in S_{\text{mejor-global}}$$

$$\tau_{rs} \leftarrow (1 - \rho) \times \tau_{rs} \quad \forall a_{rs} \in S_{\text{peor-actual}} \quad y \quad a_{rs} \notin S_{\text{mejor-global}}$$

- Se realiza una mutación de los rastros de feromona para introducir diversidad en el proceso de búsqueda. Para llevarla a cabo, el rastro de feromona asociado a cada una de las transiciones desde cada nodo (por ejemplo, cada fila de la matriz de rastros de feromona) se muta con una probabilidad P_m , utilizando cualquier operador de mutación con codificación real.

La propuesta original del SMPH aplicaba un operador que alteraba los rastros de feromona de cada transición mutada añadiendo o restando la misma cantidad en cada iteración. El rango de mutación $mut(it, \tau_{\text{umbral}})$, que depende de la media de los rastros de feromona en las transiciones de la mejor solución global, τ_{umbral} , es más suave en las primeras etapas del algoritmo -donde no hay riesgo de estancamiento- y más fuerte en las últimas etapas, donde el peligro de estancamiento es más fuerte:

$$\tau'_{rs} \leftarrow \begin{cases} \tau_{rs} + mut(it, \tau_{\text{umbral}}), & \text{si } a = 0 \\ \tau_{rs} - mut(it, \tau_{\text{umbral}}), & \text{si } a = 1 \end{cases}$$

Donde a es un valor aleatorio en $\{0, 1\}$ e it es la iteración actual.

- Como en otros modelos de OCH, el SMPH considera la reinicialización de los rastros de feromona cuando se estanca la búsqueda, lo que se lleva a cabo fijando cada rastro de feromona a τ_0 . En las primeras versiones del algoritmo se comprobaba si el porcentaje de arcos distintos entre la mejor solución y la peor de la población actual era menor que un cierto valor umbral. En versiones más recientes se utiliza un concepto diferente para evaluar si el algoritmo se ha estancado o no: se comprueba el porcentaje de iteraciones del algoritmo sin haber conseguido una mejora en la mejor solución.

2.3.4.6 PARARELIZACIÓN.

Existen numerosos estudios de este tipo de algoritmos implementados para entornos paralelos, ya que la individualidad de la hormiga se ajusta a la independencia de procesos en paralelo [BAR021]. Entre estos estudios, cabe destacar los publicados por Bullnheimer, Kotsis y Strauss [BUL97], donde exponen un modelo de implementación paralela síncrona y otra parcialmente asíncrona.

En el estudio sobre un modelo síncrono, el proceso master crea procesos hijos por cada hormiga de la comunidad, distribuyendo la información inicial necesaria para la ejecución. Cada proceso calcula una longitud del problema y lo devuelve al master que se encarga de actualizar la feromona y la gestión de los resultados.

La segunda implementación, parcialmente asíncrona, intenta optimizar la comunicación reduciéndola. Para ello, las hormigas realizan varias iteraciones del algoritmo antes de informar al master de los resultados y posteriormente proceder a la sincronización global. Una vez concluidos estos ciclos se procede, al igual que en el caso anterior, a la actualización de la feromona y gestión de los resultados por el master.

Un modelo propuesto por Barán [BAR021] establece que cada procesador realice la computación del problema para cierto número de hormigas, de la misma forma que se lleva a cabo en un contexto secuencial, obteniendo resultados parciales que al cabo de cierto número de ciclos serán transmitidos a los otros procesadores (sin ninguna sincronización), con el fin de colaborar todos en la solución del problema global, eligiendo para esto como hormigas migrantes a aquellas que hayan obtenido las mejores soluciones en su procesador.

Realiza una simulación de los algoritmos genéticos en base a políticas migratorias en los siguientes parámetros:

- *El intervalo de migración:* que establece cada cuantos ciclos cierta cantidad de hormigas migrarán de un procesador a otro.
- *La tasa de migración:* que indica cuantas hormigas han de comunicarse a los otros procesadores destino cuando se cumpla el intervalo de migración.
- *El criterio de selección:* que determina la política a seguir para la selección de las hormigas que han de migrar.

En este conjunto de algoritmos se utilizan distintos nodos de cómputo (usualmente ordenadores conectados en una red) para intentar solucionar un problema concreto. Este tipo de algoritmos tratan esencialmente de mejorar los tiempos de cómputo o bien incrementar la potencia de cálculo disponible.

Como se puede comprobar, este tipo de paralelización no ofrece, en principio, grandes cambios en la estructura de los algoritmos, simplemente distribuye las tareas de cómputo de manera más o menos eficiente entre diversos procesadores, lo que permite obtener soluciones en un tiempo menor (o realizar un mayor número de iteraciones del algoritmo en la misma cantidad de tiempo).

En este tipo de paralelización hay que prestar mucha atención a los tiempos de comunicación entre procesadores. Es bien conocido que cualquier proceso que se desee disponer en paralelo no incrementa su velocidad de manera directamente proporcional al número de nodos que se empleen, sino que el límite se encuentra en una cota bastante inferior, regida por la llamada ley de Amdahl [ALO00]. Este límite viene impuesto normalmente por los tiempos de comunicación y sincronización que existen en los algoritmos paralelos.

2.3.5 PROBLEMAS COMBINACIONALES TRATADOS.

Los algoritmos de optimización basados en colonias de hormigas se han aplicado fundamentalmente a problemas de optimización combinatoria. Dentro de este conjunto de problemas, se han utilizado en la resolución de problemas de optimización combinatoria NP-duros en los que las técnicas clásicas ofrecen resultados no muy convincentes, y problemas de caminos mínimos donde la instancia del problema varía en el tiempo.

Como ya se ha mencionado, el primer problema donde se aplicaron este tipo de algoritmos fue el de Vendedor Viajante (*Traveling Salesman Problem*) [DOR96][DOR97], instancia de un problema NP-duro, que además incluye la resolución de un camino mínimo. Este problema se ha convertido en un estándar para la realización de pruebas en modelos posteriores al inicial planteado por Dorigo.

Otros conjuntos de problemas tratados posteriormente por este tipo de algoritmos, fueron el problema de asignación cuadrática (*Quadratic Assignment Problem*) [MAN99] y el problema de ordenamiento secuencial (*Sequential Ordering Problem*) [GAM00].

Posteriormente, una de las aplicaciones donde mejor se han adaptado este tipo de algoritmos ha sido en el enrutamiento de redes (*Telecommunications Networks*), siendo los primeros trabajos desarrollados los de Schoonderwoerd y otros [SCH96] y AntNet de Di Caro y Dorigo [DOR98].

Otro tipo de problema, muy estudiado por este tipo de algoritmos, ha sido el de determinación de rutas de vehículos (*Vehicle Routing Problem*) [BUL99], problema donde los resultados de este tipo de algoritmos han sido aceptables.



Además de los anteriores ejemplos de problemas donde este tipo de algoritmos han sido probados, existen otros muchos campos de aplicación, donde los resultados están siendo prometedores como por ejemplo el diseño de circuitos lógicos [MEN01].

Recorridos de grafos mediante hormigas 3

3 RECORRIDOS DE GRAFOS MEDIANTE HORMIGAS.

En el presente proyecto, teniendo como base lo expuesto en el [punto 2](#), se intenta la realización de un estudio sobre un problema muy típico como es la búsqueda del camino más corto entre dos puntos cualesquiera en una red representada por un grafo, pero basándonos para su resolución en algoritmos de Colonias de Hormigas.

En este punto vamos a partir reflejando los objetivos marcados que tiene que cumplir el proyecto, y a partir del establecimiento de esta meta, realizar las tareas correspondientes buscando la mejor adaptación de este tipo de algoritmos al problema a resolver.

La forma de presentar los estudios realizados, resultados y experimentos va a seguir un esquema incremental. El proceso parte de una primera idea de resolución, buscando en el tiempo soluciones a los problemas encontrados y reflejando todas estas actuaciones realizadas hasta la conclusión del proyecto.

El modelo utilizado para la elaboración de las diferentes versiones a realizar en el proyecto se pensó basándose en un proceso iterativo, siguiendo cada fase la siguiente estructura:

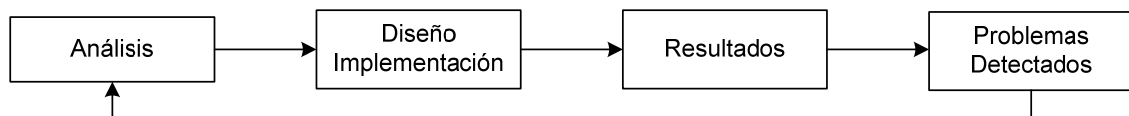


Ilustración 12 – Proceso de elaboración de cada fase del proyecto.

Para mostrar mejor el proceso realizado, se han identificado una serie de fases que comprenden modificaciones sustanciales de la solución original. Esta forma de plasmar los avances del proyecto se ha considerado más pedagógica para su comprensión y análisis de su evolución.

La denominación, de las diferentes fases expuestas, se ha intentado que describa la modificación más sustancial que ha originado la fase. De un modo escueto y de forma informativa las fases identificadas son las siguientes:

- Adaptación de [Ant System](#) al problema a resolver.
- División del problema.
- Orientación de la colonia de hormigas hacia su objetivo o meta.
- Establecimiento de una referencia única en el depósito de feromonas.

Después de informar de los objetivos se procede a desarrollar cada una de las fases de una manera detallada en los siguientes puntos.



3.1 OBJETIVOS.

Los objetivos del proyecto se centran en el análisis de los algoritmos basados en colonias de hormigas a un problema complejo como es la determinación de la distancia mínima entre dos puntos cualesquiera de una red.

El proyecto parte de algoritmos implementados para un problema similar, el del viajante de comercio. Partiendo de este desarrollo, hay que adaptar lo realizado a un nuevo modelo, que presenta similitud en el tipo de problema pero a la vez divergencias en comportamiento, originando por lo tanto una adaptación de lo anteriormente realizado, de tal manera que los resultados obtenidos sean aceptables y cualificados para informar del nivel de utilidad de estos algoritmos al problema objeto de estudio.

De una manera concreta se pueden resumir los objetivos del proyecto en los siguientes puntos.

- Comprobación de la eficiencia de los algoritmos basados en colonias de hormigas en problemas de cálculo de rutas mínimas. En otras palabras, ¿se pueden utilizar estos tipos de algoritmos para la resolución del problema a tratar?
- Adaptación y comprobación de la eficiencia de los algoritmos basados en colonia de hormigas al problema a resolver, en nuestro caso al trazado de coste mínimo de rutas de vehículos entre dos puntos cualesquiera de una red.
- Mejora del algoritmo buscando una optimización de los resultados.
- Introducción de nuevos conocimientos y adaptación de estos al problema a resolver
- Cuantificación de las dimensiones de la red capaces de ser tratadas por este tipo de algoritmos proporcionando resultados aceptables.

Estos son objetivos directos, pero también se pueden definir objetivos indirectos como son:

- Introducción de nuevos modelos de conocimientos dentro de este campo de investigación.
- Mejora de rendimientos de algoritmos utilizados actualmente en la determinación de rutas de coste mínimo.
- Despejar incógnitas respecto al rendimiento de estos modelos en este tipo de problemas.
- Establecimiento de nuevos rumbos, o simbiosis, con otros desarrollos para la mejora de resultados sobre este u otros campos.

3.2 ALGORITMO ANT SYSTEM EN EL CÁLCULO DE RUTAS DE DISTANCIA MÍNIMA ENTRE DOS PUNTOS CUALESQUIERA DE UNA RED.

En este punto de desarrollo del proyecto no se disponía de una información exhaustiva de como solucionar el problema y de las dificultades que surgirían a lo largo de la elaboración del proyecto. Se disponía de información de trabajos realizados sobre problemas similares, pero de tamaño relativamente pequeño.

Se quería hacer una similitud al problema de determinación de la mejor ruta entre dos ciudades o puntos y como funcionaba este tipo de algoritmos para la resolución de este problema. La primera pregunta que se originó era como realizar la adaptación, de este tipo de algoritmos, al problema en concreto y como introducir novedades que proporcionasen mejoras a los estudios precedentes.

La primera labor a realizar consistía en la realización de un análisis y diseño del sistema, que con las premisas y modificaciones esperadas, fuese lo suficientemente flexible para un futuro. Ya se consideró, por el tipo de problema a resolver, la necesidad futura de modificaciones y adaptaciones según los resultados de las pruebas que se fueran realizando.

Había que tener una referencia para comprobar el avance, o retroceso, que las modificaciones sobre el algoritmo original se fuesen realizando. Como punto de partida, a modo de referencia, se realizó la implementación del algoritmo [Ant System](#), [DOR961] con las modificaciones necesarias para la adaptación al problema objeto de estudio.

3.2.1 ANÁLISIS DEL SISTEMA.

Como se ha indicado anteriormente, el estudio se centra en el comportamiento de los algoritmos basados en colonia de hormigas en el cálculo de rutas mínimas entre dos puntos cualesquiera de una red. Se toma como idea, a semejanza del mundo real a representar computacionalmente, la simulación de una red de carreteras, y ver hasta que tamaño es capaz de proporcionar este tipo de algoritmos una información de utilidad. Para ello, se establecen una serie de requisitos que tiene que cumplir la solución propuesta y que a continuación se enumeran:

- La red tendrá un número determinado de puntos (ciudades) y arcos (carreteras de unión entre las diferentes ciudades).
- Un punto estará unido, a ninguno o varios puntos de la red, por arcos.
- Los arcos tendrán un valor o peso, que vendrá a representar la distancia que separa a dos puntos.
- Los arcos serán de un único sentido. El arco que une dos nodos, puede tener diferente valor de peso, según el sentido elegido.
- Se pueden elegir puntos cualesquiera de la red como origen ó destino.

Una vez identificados los requisitos de la red objeto de estudio, se procedió a la búsqueda de una solución que cumpliera los requisitos especificados del problema,

sabiendo que esta primera implementación tomaba como base para su desarrollo el modelo [Ant System](#).

En la simulación del mundo real planteada, los puntos del grafo venían a representar las diferentes ciudades, mientras que los arcos simulan las vías que los unen, siendo el valor del peso del arco una representación de la distancia entre ambas ciudades por el camino de unión.

Para la obtención de los primeros resultados y observación exhaustiva del comportamiento del algoritmo implementado, se diseñó una red de puntos sobre la que se realizarían las pruebas, de esta manera se podían verificar los resultados obtenidos mediante logs de ejecución, depurando errores que se pudieran cometer en la fase de programación.

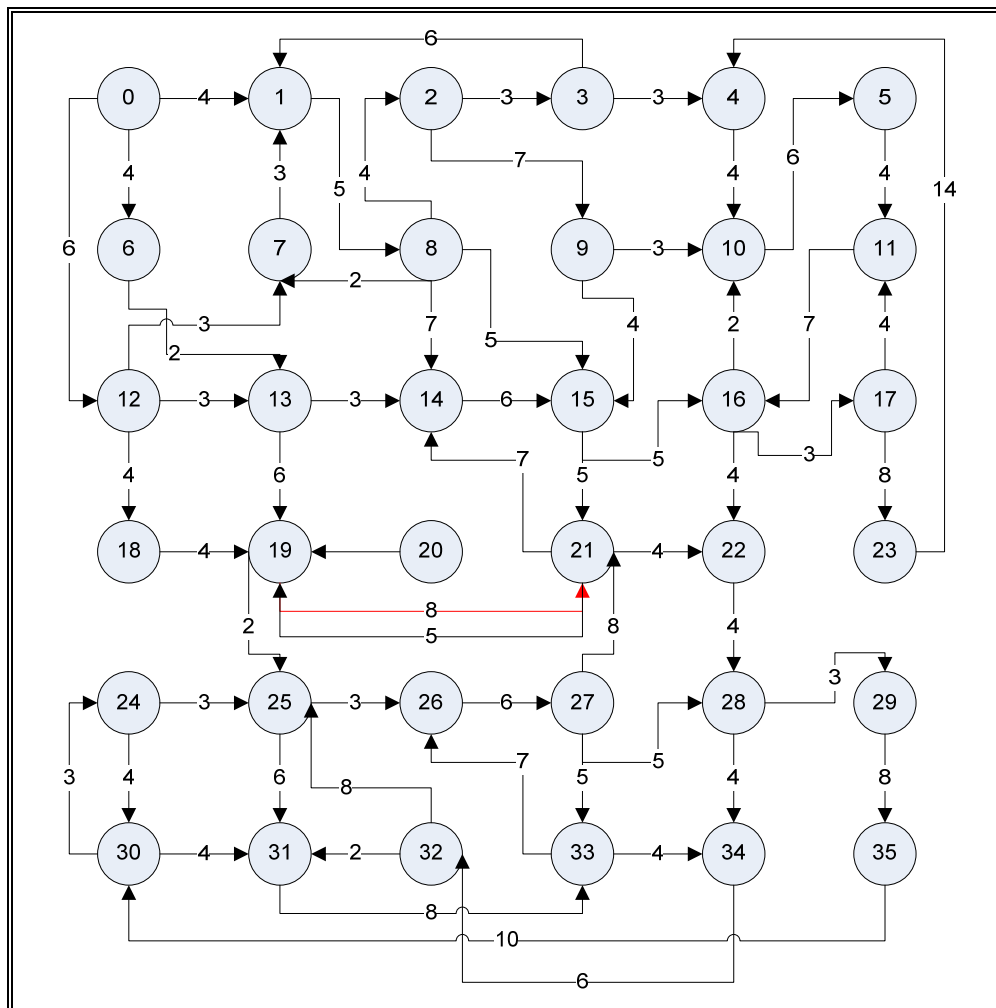


Ilustración 13 – Diseño de la red de puntos de pruebas.

Con esto quedan definidas las características del problema a resolver. A partir de estos requisitos y tomando como base el modelo [Ant System](#), se procedió a realizar la primera aproximación a la solución, la adaptación al problema concreto a resolver, sabiendo que este tipo de problemas requieren gran cantidad de pruebas y búsquedas de mejores soluciones en su ciclo de vida.



3.2.2 DISEÑO DEL SISTEMA E IMPLEMENTACIÓN.

Para realizar pruebas sobre un mismo conjunto de datos en diferentes momentos, se procedió, como primera necesidad, a la elaboración de una base de datos donde almacenar la información del estado de la red. La aplicación tendría que ser capaz de poder gestionar una red, almacenando o recuperando esta de la base de datos en cualquier momento.

Una vez completado el diseño de la base de datos, se procedió al desarrollo del software gestor de datos y ejecutor de los algoritmos que se fueran implementando. Esta ejecución ofrecería los resultados necesarios para la elaboración del proyecto y comprobación de su efectividad.

Además del almacenamiento y recuperación de la red de puntos, el software desarrollado tendría que ser capaz de generar redes completamente aleatorias, con tamaños definidos por el usuario en un momento dado. Como el objetivo principal es el estudio del rendimiento de estos algoritmos, según el tamaño de la instancia del problema, el poder elegir tanto el número de nodos, como arcos de la red, es un requisito que el software desarrollado tenía que cumplir.

Para la fase de programación se decidió la utilización de Visual Studio, y más concretamente Visual Basic, por ser un lenguaje versátil, de fácil desarrollo y que cumplía con todos los requisitos necesarios para la consecución de los objetivos.

3.2.2.1 DISEÑO DE LA BASE DE DATOS

Para poder trabajar y comprobar datos, sobre todo de comportamiento de los algoritmos utilizados, se precisaba realizar pruebas en una red conocida y que se pudiera utilizar en el tiempo, sin variaciones para la comparación de los resultados. Además, había que tener en consideración la posibilidad de almacenamiento de una red de tamaño considerable, por si en un futuro fuera necesaria su utilización.

Para almacenar esta información no se requerían muchas necesidades en cuanto a sistema de gestión datos. El modelo de datos era simple, por lo que las necesidades de gestión no eran su exigencia primordial. Como se iba a utilizar un entorno de programación de Microsoft se decidió la utilización, para cubrir esta necesidad, de Microsoft Access.

El modelo de datos diseñado, tenía que tener en cuenta variaciones futuras y la utilización de campos, que en un momento dado, pudieran aportar conocimiento a la resolución del problema.

El modelo de datos diseñado es el siguiente:

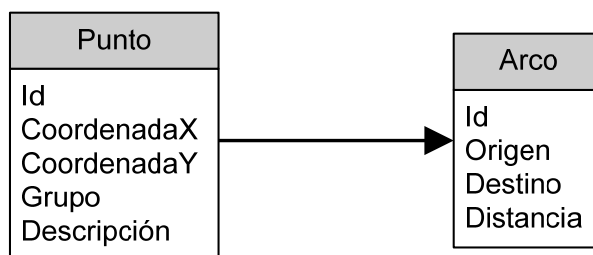


Ilustración 14 – Modelo de datos estático.

A continuación se realiza una breve descripción de los diferentes campos que conforman las tablas identificadas en esta fase de diseño:

PUNTO	
NOMBRE DEL CAMPO	DESCRIPCIÓN
ID	Identificador del punto.
COORDENADA X	Coordenada X de representación del punto en el espacio.
COORDENADA Y	Coordenada Y de representación del punto en el espacio.
GRUPO	Grupo al que pertenece el punto.
DESCRIPCIÓN	Comentarios del punto identificado.

Tabla 5 - Diseño Base de Datos. Tabla Puntos

ARCOS	
NOMBRE DEL CAMPO	DESCRIPCIÓN
ID	Identificador del arco.
ORIGEN	Punto origen del arco.
DESTINO	Punto destino del arco.
DISTANCIA	Peso del arco. En nuestro caso significa distancia entre origen y destino.

Tabla 6 - Diseño Base de Datos. Tabla Arcos.

Además de esta identificación de campos cabe mencionar la relación existente de (0, n) entre la tabla puntos y arcos.

- Un punto puede tener de ninguno o varios arcos.
- Un arco tiene que pertenecer a un punto origen.
- Un arco tiene que tener un punto destino.

Para el cumplimiento de estos requisitos, se ha procedido a establecer dos relaciones entre ambas tablas, siendo los campos origen y destino, claves ajenas de la

tabla puntos. Con el establecimiento de estas relaciones se cumplen los tres puntos anteriormente mencionados.

3.2.2.2 DISEÑO DEL SOFTWARE.

Con el entorno de desarrollo elegido, se pasa a decisiones más concretas, como puede ser la implementación del algoritmo, la estructura de datos elegida y adecuación del algoritmo tradicional al problema objeto de estudio, consiguiendo posteriormente, que este sea capaz de evolucionar implantando las mejoras oportunas que permitan una optimización de los resultados.

Este tipo de proyectos sufre variaciones constantes según los resultados obtenidos, y se va modificando en el tiempo hasta satisfacer los objetivos. Esto se ha tenido en cuenta al inicio del desarrollo del software, por lo que se ha realizado un análisis para determinar los diferentes componentes que lo conforman, intentando que las posibles modificaciones sean asequibles de realizar y la reutilización del resto del software sea un hecho.

Para ello, se han identificado un conjunto de clases que representan el modelo a desarrollar y a la vez, encapsulan las funcionalidades de cada una de ellas. En este primer estudio han surgido las siguientes clases:

- **CBasedeDatos:** Clase para gestionar las acciones de la base de datos.
- **CHormiga:** Clase para gestionar una hormiga artificial.
- **CPunto:** Clase para gestionar un punto de la red.
- **CArco:** Clase para gestionar un arco de la red.
- **CListaPuntos:** Clase para manejar el conjunto de la red, tanto los puntos como los arcos que la componen.
- **CListaHormigas:** Clase para gestionar el conjunto de hormigas que forman la colonia.

Una vez identificados los elementos, otro detalle ha tener en cuenta en esta primera aproximación es la identificación, a grandes rasgos, de las diferentes funcionalidades que debe de presentar el software desarrollado. En la siguiente ilustración se pueden observar las funcionalidades detectadas.

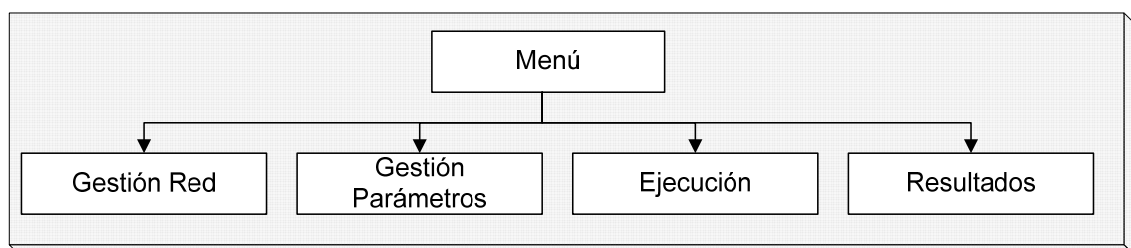


Ilustración 15 – Principales funcionalidades.

El realizar este primer estudio, teniendo en cuenta las variaciones que en un futuro puedan surgir, aportará grandes ventajas en la evolución del proyecto. Los cambios serán fáciles de identificar, tanto donde realizarlos, como las repercusiones que origina la modificación realizada.

Las funcionalidades indicadas anteriormente se pueden gestionar siguiendo el proceso que a continuación se detalla:

En el estado inicial, la única opción es la creación o recuperación de una red de puntos. Una vez se tiene en memoria la red, se procede a establecer las diversas variables consideradas en el programa, como pueden ser, el número de hormigas de la colonia, la feromona inicial depositada, los puntos origen y destino, etc. Con esto ya se tienen todas las condiciones para la aplicación del algoritmo objeto de estudio, mostrando los resultados que se consideren necesarios para los objetivos marcados.

Este proceso se observa con una mayor claridad y detalle en el siguiente gráfico:

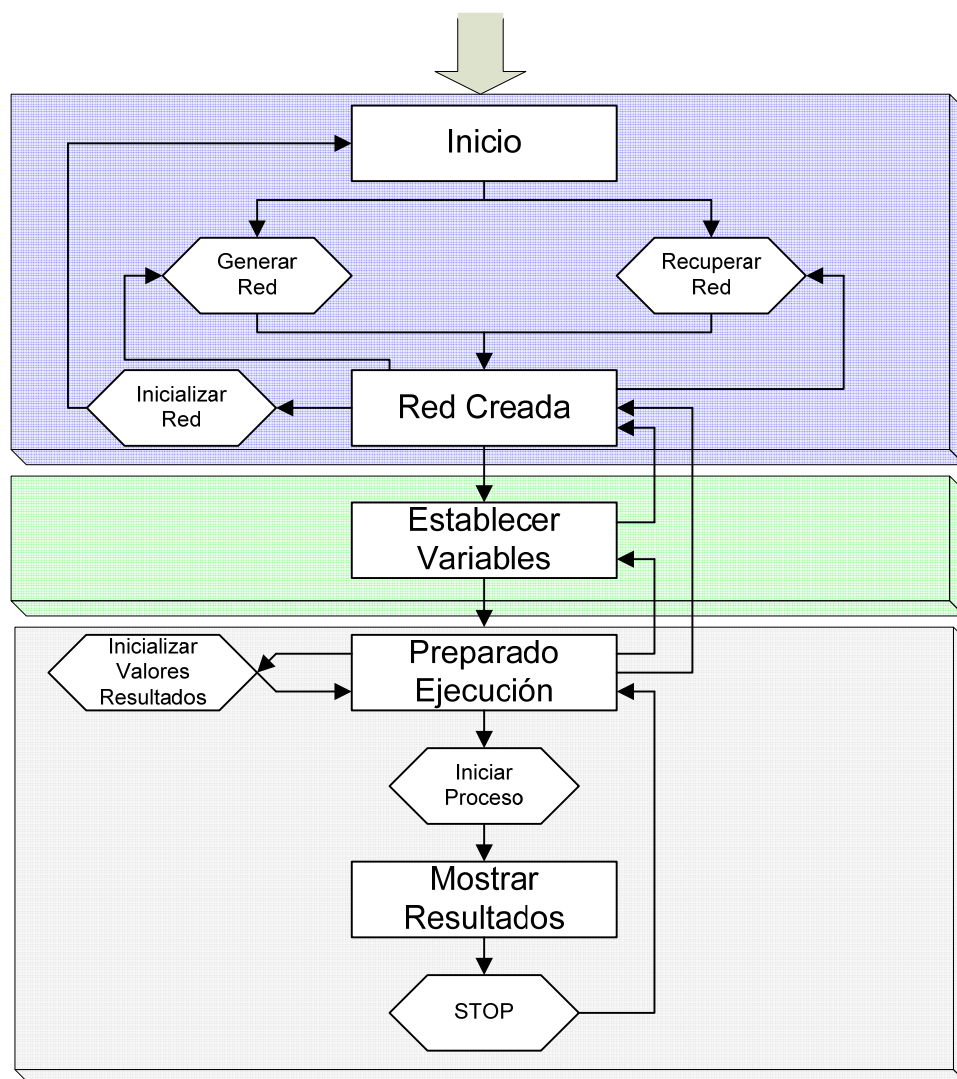


Ilustración 16 – Principales funcionalidades.

Con esto se encuentran identificados tanto los elementos que componen el software como su funcionalidad. A continuación se realiza una descripción más detallada de los diferentes elementos identificados y las funcionalidades que presenta cada uno de ellos.

3.2.2.2.1 Estructura de datos utilizada.

En el análisis del sistema se ha procedido a determinar la estructura de la red y las particularidades que hay que reflejar para transformar el problema real a una estructura computacional.

Los grafos ponderados se adaptan perfectamente al conjunto de problemas tratados por este tipo de algoritmos, y la forma de representación del grafo que mejores resultados proporciona es la [lista de adyacencias](#), al tener que reflejar grafos dispersos, originando, por consiguiente, una optimización de la memoria del sistema. Como se demostrará más adelante, esta representación del modelo satisface todos los requisitos enumerados para su resolución.

A continuación se describen los elementos de datos considerados necesarios para el estudio a realizar, como se ha representado computacionalmente y funciones necesarias, manteniendo la visión de futuro que se ha mencionado en la fase de análisis.

- **Punto**

Se corresponde con la clase CPunto identificada y simula un punto de la red. Para un punto en concreto, se han considerado una serie de campos que satisfagan las necesidades tanto de identificación, como de información para la resolución del problema.

La estructura que mejores resultados puede aportar es la que se muestra en la siguiente ilustración:

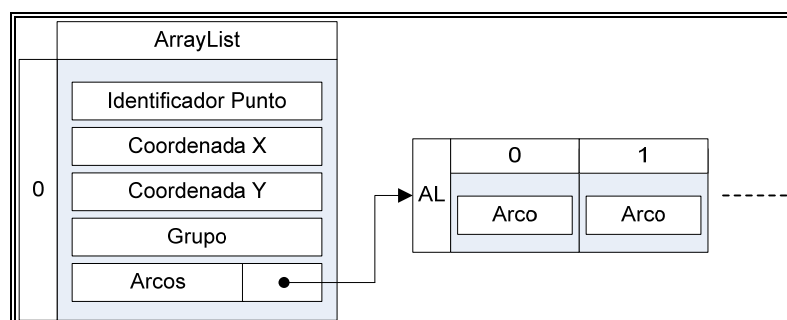


Ilustración 17 – Estructura de representación. Punto.

Las propiedades o campos identificados para esta clase son los siguientes:



CPUNTO	
PROPIEDADES	DESCRIPCIÓN
IDENTIFICADOR DE PUNTO	Identificador único de punto.
COORDENADA X	Campo de información gráfica del punto correspondiente al eje X.
COORDENADA Y	Campo de información gráfica del punto correspondiente al eje Y.
GRUPO	Campo de utilización futura para un posible agrupamiento que se pueda realizar.
ARCOS	ArrayList de elementos CArcos.

Tabla 7 – Clase CPunto. Propiedades.

Las funciones que en un primer momento, basándonos en los requisitos establecidos, se han considerado oportunas implementar son las siguientes:

CPUNTO	
FUNCIONES	DESCRIPCIÓN
AÑADIRLISTA	Se añade un arco a la lista de arcos que parten del punto.
INSERTAR	Se inserta un punto en la base de datos.
BORRARBASEDATOS	Se produce la eliminación de información de la base de datos (Puntos).
CONVTEXTO	Se convierte información del punto a línea de texto.

Tabla 8 – Clase CPunto. Funciones.

- Arco

Este elemento referencia la unión de dos puntos de la red. Debe de informar del peso o distancia que cuantifica esta unión y de los campos necesarios para simular el mundo real sobre el que se basan estos algoritmos, como es la cantidad de feromona que depositan las hormigas en su recorrido y detectan para construir la solución.

Los campos, que se han considerado oportunos reflejar, vienen identificados en la siguiente ilustración:

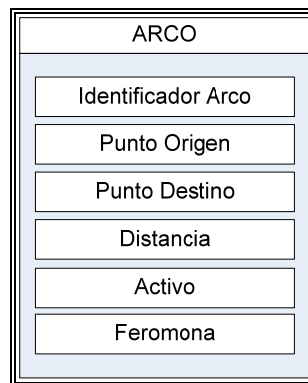


Ilustración 18 – Estructura de representación. Arco.

A continuación se realiza una breve descripción de cada campo identificado.

CARCO	
PROPIEDADES	DESCRIPCIÓN
IDENTIFICADOR DE ARCO	Identificador único de arco.
PUNTO ORIGEN	Punto de origen del arco.
PUNTO DESTINO	Punto de destino del arco.
DISTANCIA	Distancia o peso del arco.
ACTIVO	Campo que indica si el arco está en uso.
FEROMONA	Feromona depositada en el arco.

Tabla 9 – Clase CARCO. Propiedades.

En la siguiente tabla se identifican las funciones, que en un primer momento, se han considerado oportunas reflejar para el desarrollo del software.

CARCO	
FUNCIONES	DESCRIPCIÓN
BUSCARMAXID	Busca el valor de identificador más alto almacenado en la base de datos.
INSERTAR	Se inserta un arco en la base de datos.
BORRARBASEDATOS	Se produce la eliminación de información de la base de datos (Arcos).
CONVTEXTO	Se convierte información del arco a línea de texto.

Tabla 10 – Clase CARCO. Funciones.

Hay que mencionar que esta estructura de arco forma parte de un punto, es decir un punto puede contener ninguno o varios arcos. De una forma conjunta su representación sería:

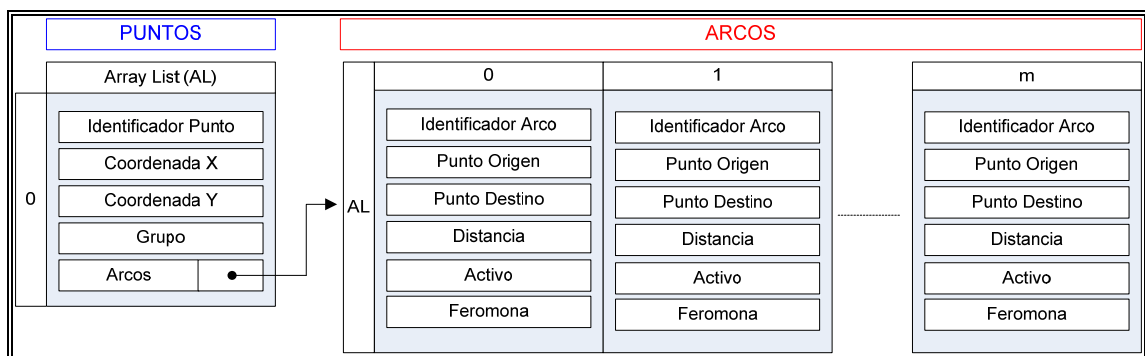


Ilustración 19 – Estructura de representación. Punto y Arcos.

- **Hormiga**

Para la representación de la hormiga se eligió una estructura similar a la representación del punto. La hormiga tenía que soportar la simulación del comportamiento de una hormiga natural, añadir las características propias de identificación de la hormiga artificial y además de prever modificaciones futuras que pudieran surgir. Con estos requisitos se decidió para su representación una estructura flexible, con una clase que gestionara sus acciones y propiedades.

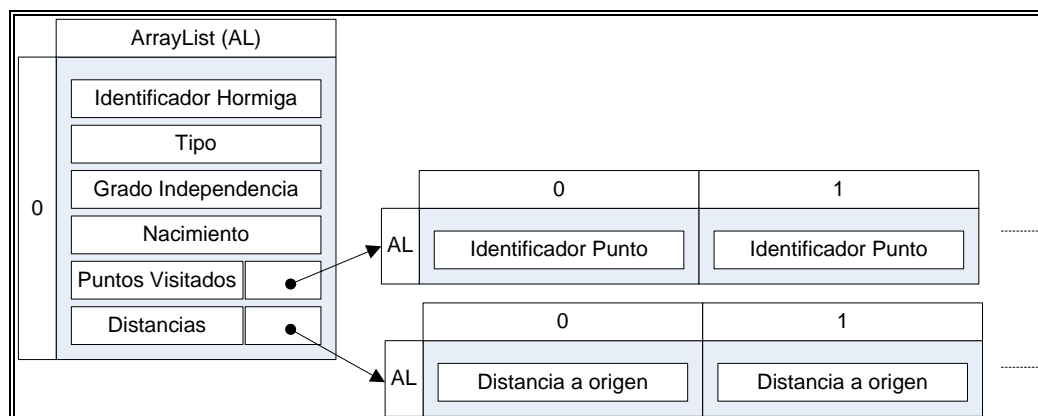


Ilustración 20 – Estructura de representación. Hormiga.

Los campos o propiedades que forman la clase son los siguientes:

CHORMIGA	
PROPIEDADES	DESCRIPCIÓN
IDENTIFICADOR DE HORMIGA	Identificador único de la hormiga de origen computacional.
TIPO	Representa las funciones que realiza la hormiga dentro de la colonia. Algunos trabajos en este campo poseen una característica similar para representar diferentes funciones de las hormigas en la comunidad.
GRADO INDEPENDENCIA	Este dato se ha introducido con vistas futuras para diferenciar a las hormigas según su fiabilidad en el seguimiento de la feromona detectada.



CHORMIGA	
PROPIEDADES	DESCRIPCIÓN
NACIMIENTO	Dato que representa la vida de una hormiga. El objeto del proyecto es realizar un estudio del comportamiento de este tipo de algoritmos en redes grandes. Esto puede originar que las hormigas se pierdan por la red y un campo de este tipo puede ayudar en detectar este problema.
PUNTOS VISITADOS	ArrayList que representa la lista tabú de la hormiga. Informa de las ciudades recorridas. <ul style="list-style-type: none">Identificador de punto: Identificador de nodo recorrido.
DISTANCIAS	ArrayList que complementa la lista tabú de puntos visitados, donde se va almacenado las distancias de los distintos puntos recorridos al origen. <ul style="list-style-type: none">Distancia a origen: Distancia que presenta al punto origen de la hormiga.

Tabla 11 – Clase CHormiga. Propiedades

Las funciones identificadas para la clase CHormiga en un primer momento son las siguientes:

CHORMIGA	
FUNCIONES	DESCRIPCIÓN
AÑADIRCIUADADISTANCIA	Movimiento de la hormiga a una nueva posición.
POSICIONESRECORRIDAS	Recorrido realizado por una hormiga.
NACIMIENTO NUEVA HORMIGA	Inicializar los valores de una Hormiga artificial ya creada anteriormente.
LOG	Escritura de información de una Hormiga a fichero de texto.

Tabla 12 – Clase CHormiga Funciones.

- Lista Hormigas**

Para la resolución de este tipo de problemas se necesita una colonia de hormigas que simule el comportamiento de la colonia real. Para ello se crea una estructura de orden superior, que englobe a todas las hormigas que en un momento dado cooperan para la resolución del problema.

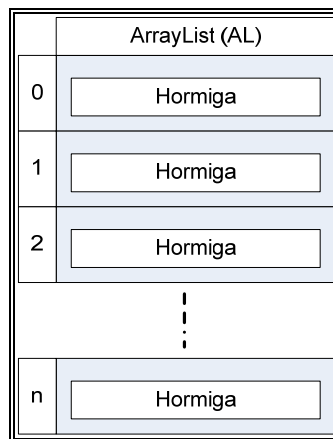


Ilustración 21 – Estructura de representación. Colonia de Hormigas.

Las propiedades y funciones de este elemento se describen en las siguientes tablas:

CLISTA HORMIGA	
PROPIEDADES	DESCRIPCIÓN
LISTA HORMIGAS	ArrayList que representa la lista de hormigas de la colonia creada.

Tabla 13 – Clase CListaHormiga. Propiedades.

CLISTA HORMIGA	
FUNCIONES	DESCRIPCIÓN
NELEMENTOS	Número de elementos, hormigas dadas de alta.
AÑADIR HORMIGA	Añade una hormiga a la lista.
NARCOS RECORRIDOS	Devuelve el número de arcos de un determinado elemento de la lista.
DEVUELVE ELEMENTO	Devuelve un elemento de la lista.
VACIAR LISTA	Vacía la lista.

Tabla 14 – Clase CListaHormiga. Funciones.

- **Lista Puntos**

Para una gestión de la red más dinámica se ha considerado oportuno introducir una estructura superior que englobe todos los puntos que forman la red objeto de estudio. Su estructura es similar a la lista de hormigas expuesta anteriormente.

Las propiedades y funciones de este elemento se describen en las siguientes tablas:



CLISTAPUNTOS	
PROPIEDADES	DESCRIPCIÓN
LISTAPUNTOS	ArrayList que representa la lista de puntos que forman la red.

Tabla 15 – Clase CListaPuntos. Propiedades.

CLISTAPUNTOS	
FUNCIONES	DESCRIPCIÓN
NELEMENTOS	Número de puntos de la red.
AÑADIRPUNTO	Añade un punto a la lista.
DEVUELTOPUNTO	Devuelve un punto determinado.
VACIARLISTA	Vacía la lista.

Tabla 16 – Clase CListaPuntos. Funciones.

3.2.2.2 Algoritmo implementado.

El algoritmo diseñado presenta algunas modificaciones del original que se ha tomado como base, la razón es que el problema a resolver cambia en algunos aspectos con relación al problema del viajante de comercio, para el que se diseñó [Ant System](#). Una de las características que definen a una metaheurística debe de ser su versatilidad, como adaptación mediante pequeños cambios a problemas diferentes. Las modificaciones introducidas son las siguientes:

- **Hormiga fiel:** Introducción de una hormiga que al final de cada ciclo recorre los arcos con mayor aporte de feromona, es decir el recorrido más probable que las hormigas utilicen. El motivo por el que se tomo en consideración esta modificación fue la comprobación de coincidencia del mejor recorrido hasta el momento identificado, con el recorrido de mayor probabilidad. Determina el estancamiento en valores no óptimos del resultado.
- **Recorrido de la red.** El algoritmo original recorría todos los puntos existentes de la red, es un requisito del problema del vendedor viajante. Para el problema actual esto no es necesario, el movimiento de una hormiga finaliza cuando llega al destino, o su movimiento se ve impedido por no tener donde desplazarse. Cuando esto ocurre se inicializa la hormiga en el punto de origen. Esto ocasiona que cada hormiga pueda presentar un número de ciudades visitadas diferentes y finalizaciones en ciclos diversos.
- **Eliminación de la heurística** basada en la distancia al punto siguiente. En el tipo de problema objeto de estudio del proyecto, el conocimiento de las distancias a los siguientes puntos no proporciona ninguna información heurística. Puede que una distancia grande acerque mucho más al objetivo

que la corta, o al revés también puede ocurrir, es esta la causa que origina la nula aportación de conocimiento a la resolución del problema tal como lo trata el algoritmo original.

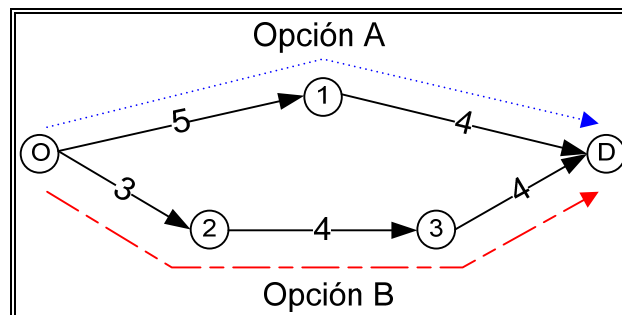


Ilustración 22 – Heurística en el algoritmo planteado.

En el punto origen O, cuando se decide que camino hay que tomar, no tiene porque tener mayor probabilidad la opción B, aunque tenga la ciudad siguiente más cercana. En este tipo de problema, de cálculo de rutas hacia un destino, lo importante es la distancia final no las intermedias.

- La **evaporación** de la feromona a posteriori, en el problema original se realizaba una vez terminada la ruta completa de las diferentes hormigas, es decir, se realizaba la deposición y la evaporación de feromona. En nuestro caso esto sufre una variación ya que una hormiga pueden realizar su recorrido en 5 ciclos y otra iniciada en el mismo momento en 30. Se ha decidido emular la evaporación lo más similar al mundo natural, por lo que se realiza en una pequeña proporción cada ciclo o movimiento de las hormigas, manteniendo un valor mínimo de feromona en cada arco. Este procedimiento también supone mayor consumo de recursos.

En la siguiente ilustración se hace una descripción simbólica de los pasos más relevantes del algoritmo implementado continuando con una breve descripción.

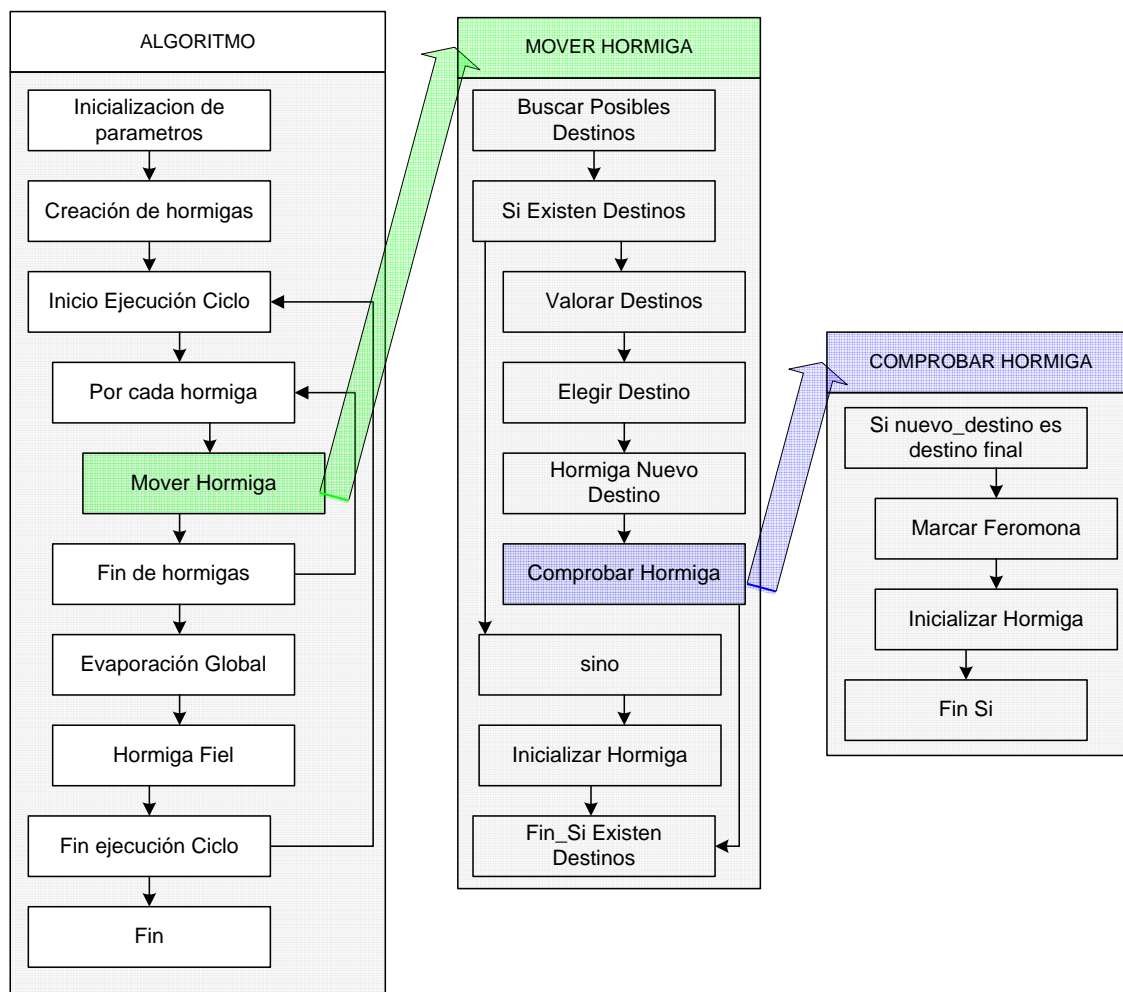


Ilustración 23 – Representación gráfica del algoritmo.

A continuación se hace una exposición textual de las funciones más representativas del algoritmo implementado.

ALGORITMO

Algoritmo que simula el comportamiento de una colonia de hormigas para la búsqueda del camino mínimo entre dos puntos cualesquiera de una red.

1. Inicializar parámetros: Se inicializan todos los parámetros necesarios según las variables introducidas por el usuario.
2. Creación de hormigas: Se realiza la creación de la colonia de hormigas que realizarán la búsqueda de la solución.
3. Inicio Ejecución Ciclo: Se inicia el proceso de búsqueda.
4. Por cada hormiga: Se recorren las hormigas creadas en el punto 2 para la ejecución de:

5. MOVER HORMIGA

6. Fin Por cada hormiga: Finalización del recorrido de las hormigas de la colonia
7. Evaporación Global: Se realiza la evaporación global de la feromona de la red. La elección de la cantidad de feromona evaporada en cada ciclo es un tanto por ciento de la existente en el arco. Dado que en cada ciclo se evapora una cantidad y estos ciclos no tiene relación con los éxitos, la cantidad elegida después de la realización de varias pruebas es de un 1% de la cantidad existente en el momento. Siempre existirá una cantidad mínima en cada arco, manteniendo posibilidades de exploración. La cantidad de feromona que mantendrá todo arco es un 10% de la feromona que se considera como inicial de todo arco.
$$feromona = \begin{cases} si & feromona > feromona_{inicial} \times 0.1 \Rightarrow feromona \times 0.01 \\ si & feromona \leq feromona_{inicial} \times 0.1 \Rightarrow feromona \end{cases}$$
8. Hormiga fiel: Cada ciclo, una hormiga (“hormiga fiel”), recorre los puntos entre el origen y destino, siguiendo siempre los arcos con mayor cantidad de feromona. Sería la ruta más probable de seguir por el conjunto de la colonia. Este proceso se inicia tras el primer éxito de alguna hormiga de la colonia.
9. Fin ejecución ciclo: Finalización de un ciclo de procesamiento.
10. Fin algoritmo: Comprobación de la finalización del algoritmo. Si no se cumple ir al punto 3.

MOVER HORMIGA

Se realiza el movimiento de una determinada hormiga de la colonia.

- 5.1 Buscar posibles destinos: Desde la posición actual de la hormiga, se buscan los destinos posibles. Existe la restricción de que una hormiga no puede visitar la misma ciudad dos veces.
- 5.2 Si existen destinos: Condición que se cumple si la hormiga tiene al menos un posible destino donde desplazarse.
- 5.3 Valorar destinos: Realiza la valoración de los posibles destinos, basándose en el depósito de feromona detectado en cada uno, siguiendo la expresión basada en [Ant System](#).

$$p_{rs}^k = \begin{cases} \frac{[r_{rs}]^\alpha}{\sum_{u \in N_r^k} [r_{ru}]^\alpha} \\ 0 & \text{No posibles} \end{cases}$$



5.4 Elegir destino: Una vez valorados los destinos en el punto anterior, se realiza la elección de manera probabilística según las valoraciones de cada posibilidad.

5.5 Hormiga en nuevo destino: Elegido el destino se procede a situar a la hormiga en su nueva posición.

5.6 **COMPROBAR HORMIGA**.

5.7 Sino: Condición que se cumple cuando la hormiga no presenta ningún posible destino

5.8 Inicializar Hormiga: Se inicializan los valores de la hormiga situándola en el punto origen.

5.9 Fin si: Finalización del movimiento de la hormiga

COMPROBAR HORMIGA

Comprueba que la hormiga ha llegado al destino final y en caso afirmativo realiza las tareas oportunas

5.6.1 Si nuevo destino es destino final: Si la hormiga ha llegado a su destino final

5.6.2 Marcar feromona: Marca con feromona el recorrido seguido por la hormiga en función de la distancia entre origen y destino. El valor se corresponde con el inverso de la distancia recorrida. A recorridos más largos, le corresponde menor cantidad de feromona, si por el contrario la distancia disminuye, el valor de la feromona depositada aumenta. Una representación más formal sería la siguiente.

$$feromona_{Arco-i} = feromona_{Arco-i} + \frac{1}{distancia_{inicio-fin}}$$

5.6.3 Inicializar hormiga: Se sitúa a la hormiga en el punto origen y se inicializan sus valores

5.6.4 Fin si:

3.2.3 RESULTADOS.

Este primer algoritmo diseñado representa un punto de partida y sobre todo una introducción al estudio del comportamiento de este tipo de algoritmos. Se han detectado una serie de problemas, originando adaptaciones en el algoritmo para representar el problema objeto de estudio, y más que los resultados en sí, la interpretación de estos ha sido lo más importante de esta primera aproximación.



Se representan tres modelos experimentales que servirán en un futuro para comprobaciones de los avances conseguidos.

Los resultados que se muestran son para tanto para la **“hormiga fiel”** (aquella hormiga que una vez completado el ciclo de las hormigas de la colonia realiza un recorrido por los arcos que poseen el mayor valor de feromona depositada, teóricamente el camino con mayor probabilidad que siguen las hormigas de la colonia), como el indicado como **“mejor resultado”**, que se refiere al mejor resultado obtenido por alguna hormiga de la colonia. Para cada una de ellas se informa del ciclo, tiempo (milisegundos) y distancia donde se ha producido el evento de éxito, tanto el primero como si este ha mejorado después de transcurrido un tiempo de 60 segundos.

En los resultados de los tres experimentos realizados se añade el resultado óptimo calculado mediante el Algoritmo de Dijkstra, que sirve para ver la eficacia de los diferentes ensayos.

La nomenclatura de identificación de las pruebas que se van a realizar durante la elaboración del proyecto corresponden a la siguiente estructura:

Experimento: **X_RN-Y-H**

- X, corresponde a un número romano que indica la fase del proyecto. (I-IV)
- R, carácter fijo de información de resultados.
- N, número de puntos de la red donde se realiza el experimento.
- Y, número de arcos por punto
- H, número de hormigas del experimento

Ejemplo: Experimento: I_R100-2-50

Indica que los resultados que se exponen corresponden a una prueba perteneciente a la fase I del proyecto, con un entorno de 100 puntos en la red, dos arcos por punto y una colonia de hormigas formada por 50 individuos.

Los resultados de los modelos elegidos son los siguientes:



• Resultados Modelo. I_R100-2-50.

ENTORNO EXPERIMENTAL	
•	Se genera una red de 100 puntos.
•	Cada punto tiene como origen 2 arcos y como destino de cada arco un punto aleatorio de la red.
•	Se crean 50 hormigas situadas en el punto de inicio.
•	El valor de feromona inicial es de 0,1.
•	Evaporación 1%.

Tabla 17 – Entorno experimental. I_R100-2-50

Puntos en la red		100					
		Hormiga Fiel			Mejor Resultado (Colonia)		
Nº Arcos por punto		2			2		
		Ciclo	Tiempo	Distancia	Ciclo	Tiempo	Distancia
	1 ^{er} Éxito	10	562	52	10	562	52
	Fin	1607	60000	52	42	2156	44

Optimo	38
--------	----

Ilustración 24 – Resultados. I_R100-2-50.

* Mejor Resultado = Mejor resultado proporcionado por una hormiga de la colonia.

• Resultados Modelo. I_R100-50-50.

ENTORNO EXPERIMENTAL	
•	Se genera una red de 100 puntos.
•	Cada punto tiene como origen 50 arcos y como destino de cada arco un punto aleatorio de la red.
•	Se crean 50 hormigas situadas en el punto de inicio.
•	El valor de feromona inicial es de 0,1.
•	Evaporación 1%.

Tabla 18 – Entorno Experimental. I_R100-50-50.

Puntos en la red		100					
		Hormiga Fiel			Mejor Resultado (Colonia)		
Nº Arcos por punto		50			50		
		Ciclo	Tiempo	Distancia	Ciclo	Tiempo	Distancia
	1 ^{er} Éxito	3	344	7	3	344	7
	Fin	1102	60000	7	1102	60000	7

Optimo	4
--------	---

Ilustración 25 – Resultados. I_R100-50-50.



* Mejor Resultado = Mejor resultado proporcionado por una hormiga de la colonia.

• Resultados Modelo. I_R100-100-50.

ENTORNO EXPERIMENTAL	
•	Se genera una red de 100 puntos.
•	Cada punto tiene como origen 100 arcos y como destino de cada arco un punto aleatorio de la red.
•	Se crean 50 hormigas situadas en el punto de inicio.
•	El valor de feromona inicial es de 0,1.
•	Evaporación 1%.

Tabla 19 – Entorno Experimental. I_R100-100-50

Puntos en la red		100					
		Hormiga Fiel			Mejor Resultado (Colonia)		
Nº Arcos por punto		100			100		
		Ciclo	Tiempo	Distancia	Ciclo	Tiempo	Distancia
	1 ^{er} Éxito	4	532	6	4	532	6
	Fin	941	60000	6	941	60000	6

Optimo	4
--------	---

Ilustración 26 – Resultados. I_R100-100-50.

* Mejor Resultado = Mejor resultado proporcionado por una hormiga de la colonia.

Como se ha mencionado anteriormente los resultados corresponden a los mejores recorridos conseguidos por la “*hormiga fiel*” (hormiga que sigue la ruta donde la cantidad de feromona depositada es mayor) y la denominada “*mejor resultado*” (corresponde su información a un individuo de la colonia), indicando ciclo y tiempo en milisegundos en que se ha producido el hecho, así como la distancia conseguida.

○ **Conclusiones.**

Son unos resultados que se aproximan al valor óptimo, pero los descubrimientos son siempre en el inicio del proceso, no evolucionando en el transcurso de los 60 segundos que se han establecido de duración del proceso de pruebas.

Cabe destacar como, en el experimento I_R100-2-50, el mejor resultado lo proporciona una hormiga de la colonia, no así la hormiga fiel. Esto quiere decir que de momento la ruta más probable no presenta la mejor opción, informando por consiguiente, de la existencia de un óptimo local del que por el momento la colonia no ha conseguido salir, desde los 2 seg., donde se ha realizado el descubrimiento, hasta los 60 seg. de duración del proceso.

Estos resultados informan de la capacidad que tiene este tipo de algoritmos para la resolución del problema a tratar, cumpliendo uno de los objetivos marcados, pero no llega a evolucionar en el tiempo lo suficiente para la consecución de mejoras en el resultado.



3.2.4 PROBLEMAS ENCONTRADOS.

Esta primera aproximación refleja la dificultad que presenta este tipo de algoritmos, no sólo en su adaptación al problema en concreto, sino también se identifican errores que en futuras versiones se tendrán que subsanar.

La evaporación de feromona resulta bastante inefectiva para este tipo de problema. La evaporación natural se realiza sobre todo el entorno y en todo momento. Esto llevado a la red objeto de estudio, origina la aplicación de una operación costosa computacionalmente, ya que acarrea numerosas operaciones y comprobaciones con lo que creamos un algoritmo ineficiente. Si por el contrario, sólo aplicamos la evaporación a la ruta correspondiente a la hormiga con un recorrido de éxito, lo que estamos haciendo es simplemente una deposición de feromona con una cantidad menor a la original, con lo que el concepto de evaporación no tiene ningún significado.

Como se ha mencionado anteriormente, esta versión es un punto de partida. Desde este momento es cuando se van a introducir modificaciones que busquen algoritmos más eficientes para la resolución del problema, sirviendo estos resultados de comprobación de los avances realizados, si es que se producen. Esta evolución es la que se reflejará en los siguientes puntos del proyecto.



3.3 DIVISIÓN DEL OBJETIVO EN ALGORITMOS BASADOS EN COLONIAS DE HORMIGAS PARA EL CÁLCULO DE DISTANCIAS MÍNIMAS.

Esta es la primera fase donde realmente se introduce nuevo conocimiento al algoritmo. Se comprobarán si estas modificaciones surten efectos positivos y efectúan una mejora de los resultados obtenidos en la fase anterior.

3.3.1 MODIFICACIONES PLANTEADAS.

Se pretende realizar una innovación bastante profunda con respecto a lo documentado sobre este tipo de problemas hasta el momento. Se sabe que la división de un problema puede resultar beneficiosa para su resolución global, es por ello, por lo que se intenta plasmar esta lógica en el algoritmo utilizando para ello dos tipos de hormigas, unas que buscan el origen y otras el destino. Esto significa que el problema teóricamente se puede dividir en dos y, por lo tanto, los tiempos en búsqueda de la solución también. Las principales características, que presenta esta primera idea de resolución del problema, son las siguientes:

- Se van a tener dos tipos de hormigas. Unas buscarán el origen de la ruta y otras el punto destino.

Rojas: Hormigas que buscan el origen de la ruta

Verdes: Hormigas que buscan el destino de la ruta.

- Cuando encuentran su objetivo depositarán feromona en la ruta seguida, existiendo una feromona distinta para cada tipo de hormiga. El depósito de feromona será función de la distancia recorrida.
- Las hormigas se dividirán en igual número entre los dos conjuntos, y cada hormiga se situará en un punto aleatorio de la red.
- Las hormigas no pueden visitar un punto más de una vez. Aquellas que no puedan moverse y continuar su ruta, serán consideradas como muertas. Por cada muerte nacerá una nueva hormiga, con las mismas características que su predecesora, en un lugar aleatorio de la red.
- El movimiento de la hormiga al siguiente punto será, dentro de sus posibles rutas, proporcional a la feromona de su tipo detectada en estos caminos.
- En la evolución del sistema se marcan los arcos con una cantidad de feromona roja o verde. Cada arco presentará una cantidad de feromona de cada clase, que puede hacer referencia al grado de proximidad que tiene a los puntos origen y destino. Esto origina que los puntos puedan ser clasificados como rojos o verdes, dependiendo de la cantidad de feromona de cada tipo que contengan sus arcos.

- Cada cierto tiempo se puede lanzar la búsqueda de la solución, siguiendo si es nodo rojo, la feromona roja y si es nodo verde, la feromona verde. Nodo rojo se establece aquel que tiene mayor cantidad, de media por arco de salida, de feromona roja que de feromona verde. Se considera nodo verde en caso contrario.

Esta primera idea parecía un comienzo alentador, pero un análisis más detallado hizo aflorar problemas.

A continuación se detallan las modificaciones introducidas en el modelo para poder ajustar el algoritmo a los nuevos requisitos.

3.3.2 MODIFICACIONES REALIZADAS.

3.3.2.1 DISEÑO DE ARCOS Y PUNTOS

El primer problema detectado es el modelo diseñado para representar la red. La red diseñada para representar el problema no es bidireccional, el peso del arco que une los nodos i y j es $W_{ij} = 5$, por el contrario, cuando el sentido es inverso el valor puede ser diferente, $W_{ji} = 6$. En este nuevo diseño, las hormigas rojas en la búsqueda del origen, realizan la búsqueda en sentido contrario, originando un nuevo requisito que había que tenerlo en cuenta en la nueva implementación.

Se realizó un nuevo diseño de puntos y arcos que se indica en la siguiente ilustración.

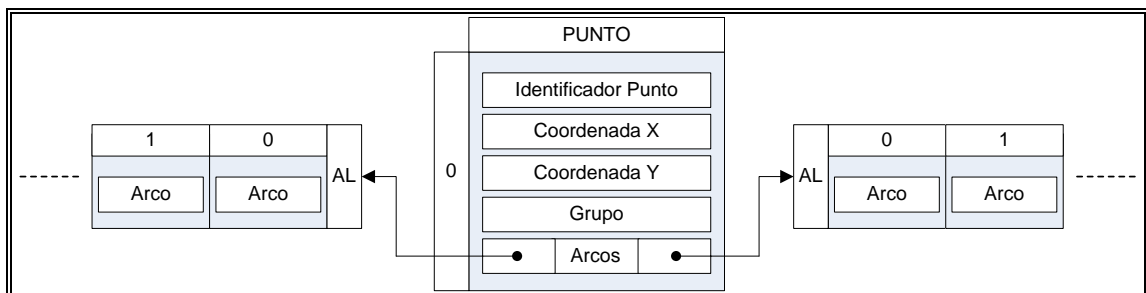


Ilustración 27 – Representación de Punto. Modelo Bidireccional.

Se estableció la bidireccionalidad de los puntos creando dos estructuras de arcos en cada punto. Una representaba los arcos donde el punto era origen y otra donde era destino.

Esta nueva disposición, permitía recorrer la red en ambos sentidos desde un punto, dando la posibilidad de ir en busca del origen solicitado, función que el modelo original no contemplaba.

Una vez solucionado este problema había que poder clasificar los puntos en un momento determinado, utilizando para ello el campo grupo. Dependiendo de la cantidad de feromona, tanto roja como verde que contienen los arcos, se puede clasificar un

punto y en consecuencia indicar teóricamente de donde se encuentra más cercano, si del origen o del destino.

En el diseño del arco hay que añadir otro campo feromona. Un arco tiene que poder representar la cantidad de feromona de cada clase que contiene, y estos valores son los que clasifican, según su valoración, el tipo de punto dentro del modelo diseñado.

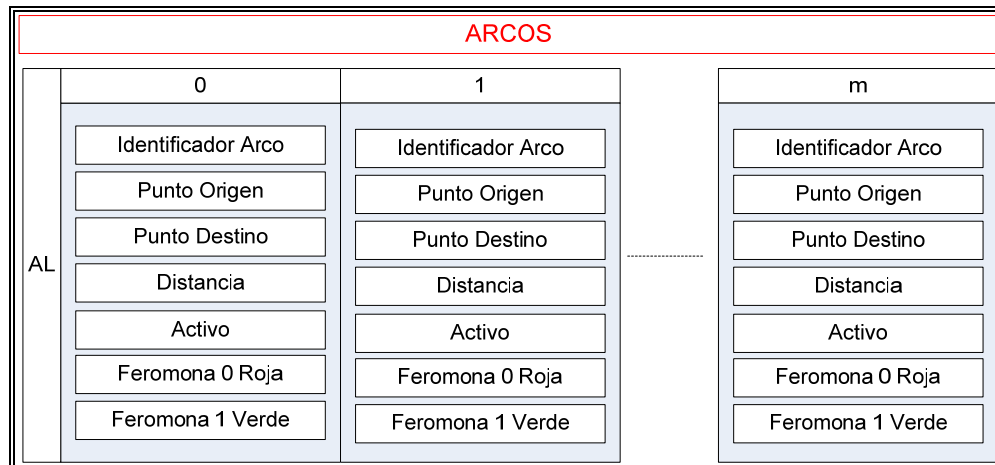


Ilustración 28 – Diseño Arcos.

3.3.2.2 MODIFICACIONES REALIZADAS EN EL ALGORITMO

El algoritmo sufre unas modificaciones que a continuación se enumeran:

- Creación de dos tipos de hormigas: Se definen en la inicialización dos conjuntos de hormigas de igual magnitud.
- Eliminación de la evaporación: Como se ha mencionado en el [punto 3.2.4](#) anteriormente expuesto, la evaporación origina un problema en cuanto a consumo computacional y su influencia en los resultados no es tan destacada, es por ello por lo que se decide la no utilización de este concepto.
- Situación aleatoria en el nacimiento de las hormigas: Se va a distribuir la colonia de hormigas existentes de forma aleatoria en la red de puntos.
- Depósito de feromona: Cuando se realiza el depósito de feromona sobre un arco determinado, esta deposición puede ser, según el tipo de hormiga, roja o verde. Como se ha mencionado anteriormente, los puntos de la red se van a clasificar según la cantidad de feromona que contengan de cada tipo. Para poder realizar esta clasificación se actúa de la siguiente manera.

$$Hormiga_{Origen-Roja} \Rightarrow feromona_{Arcoi \rightarrow j(Roja)} = feromona_{Arcoi \rightarrow j(Roja)} + \frac{1}{Distancia_{recorrido}}$$

$$Hormiga_{Destino-Verde} \Rightarrow feromona_{Arcoi \rightarrow j(Verde)} = feromona_{Arcoi \rightarrow j(Verde)} + \frac{1}{Distancia_{recorrido}}$$



El campo tipo de un punto va a ser un campo numérico inicializado a 0.

- Una hormiga que en el recorrido hacia el origen ha pasado por el punto P, le corresponde un depósito de feromona roja con un valor x . A la vez que se actualiza el arco correspondiente, también se añade esa cantidad con valor positivo al campo tipo.
- Si por el contrario la hormiga que ha pasado por el punto P corresponde a una hormiga que busca el destino y por lo tanto realiza la deposición de feromona verde, al actualizar el arco correspondiente también actualiza el campo tipo, pero esta vez con valor negativo.
- Esto origina la clasificación de cada punto en un grupo según el valor de tipo. Si el valor es positivo, el punto está teóricamente más cercano al origen, si por el contrario es negativo, está más cercano al destino. En el siguiente ejemplo se puede comprobar la evolución en el tiempo más nítidamente:

La feromona en cada arco tiene un único valor que corresponde a la suma de la feromona inicial que presentan todos los arcos más la feromona depositada por las hormigas de la colonia durante la ejecución del algoritmo. En el modelo gráfico se han separado las dos clases de feromona para una mejor comprensión. La evolución representa el estado de un punto, en cuatro momentos diferentes, durante la ejecución del algoritmo.

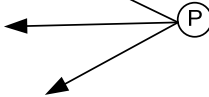
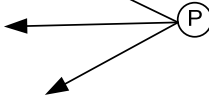


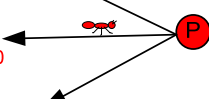
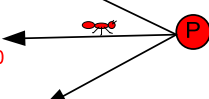
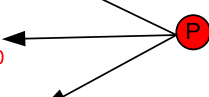
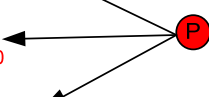
EVOLUCIÓN DEL DEPÓSITO DE FEROMONA				
Estado Inicial	Feromona roja	Arcos de búsqueda del Origen	Arcos de búsqueda del Destino	Feromona verde
	Feromona inicial : 0,001 Feromona depositada : 0			Feromona inicial : 0,001 Feromona depositada : 0
	Feromona inicial : 0,001 Feromona depositada : 0			Feromona inicial : 0,001 Feromona depositada : 0
	Feromona inicial : 0,001 Feromona depositada : 0			Feromona inicial : 0,001 Feromona depositada : 0
	Feromona inicial : 0,001 Feromona depositada : 0			Feromona inicial : 0,001 Feromona depositada : 0
		Tipo : 0.000		
$Tipo = \left(\frac{(0,001+0)+(0,001+0)+(0,001+0)}{3} \right) - \left(\frac{(0,001+0)+(0,001+0)}{2} \right)$				
Ciclo n	Feromona roja	Arcos de búsqueda del Origen	Arcos de búsqueda del Destino	Feromona verde
	Feromona inicial : 0,001 Feromona depositada : 0			Feromona inicial : 0,001 Feromona depositada : 0,0011
	Feromona inicial : 0,001 Feromona depositada : 0			Feromona inicial : 0,001 Feromona depositada : 0
	Feromona inicial : 0,001 Feromona depositada : 0			Feromona inicial : 0,001 Feromona depositada : 0
	Feromona inicial : 0,001 Feromona depositada : 0			Feromona inicial : 0,001 Feromona depositada : 0
		Tipo : (0,001) - (0,00155) = - 0,00055		
$Tipo = \left(\frac{(0,001+0)+(0,001+0)+(0,001+0)}{3} \right) - \left(\frac{(0,001+0,0011)+(0,001+0)}{2} \right)$				
Ciclo n+p	Feromona roja	Arcos de búsqueda del Origen	Arcos de búsqueda del Destino	Feromona verde
	Feromona inicial : 0,001 Feromona depositada : 0			Feromona inicial : 0,001 Feromona depositada : 0,0011
	Feromona inicial : 0,001 Feromona depositada : 0,010			Feromona inicial : 0,001 Feromona depositada : 0
	Feromona inicial : 0,001 Feromona depositada : 0			Feromona inicial : 0,001 Feromona depositada : 0
	Feromona inicial : 0,001 Feromona depositada : 0			Feromona inicial : 0,001 Feromona depositada : 0
		Tipo : (0,00433) - (0,00155) = 0,00278		
$Tipo = \left(\frac{(0,001+0)+(0,001+0,010)+(0,001+0)}{3} \right) - \left(\frac{(0,001+0,0011)+(0,001+0)}{2} \right)$				
Ciclo n+p+q	Feromona roja	Arcos de búsqueda del Origen	Arcos de búsqueda del Destino	Feromona verde
	Feromona inicial : 0,001 Feromona depositada : 0,007			Feromona inicial : 0,001 Feromona depositada : 0,0011
	Feromona inicial : 0,001 Feromona depositada : 0,010			Feromona inicial : 0,001 Feromona depositada : 0
	Feromona inicial : 0,001 Feromona depositada : 0			Feromona inicial : 0,001 Feromona depositada : 0
	Feromona inicial : 0,001 Feromona depositada : 0			Feromona inicial : 0,001 Feromona depositada : 0
		Tipo : (0,00667) - (0,00155) = 0,00512		
$Tipo = \left(\frac{(0,001+0,007)+(0,001+0,010)+(0,001+0)}{3} \right) - \left(\frac{(0,001+0,0011)+(0,001+0)}{2} \right)$				

Tabla 20 – Evolución puntos. Clasificación.

La feromona representada en el campo tipo, de un punto cualquiera de la red, vendría expresada por la siguiente formula:

$$feromona_{Punto_i} = \frac{\sum_1^{n^{\circ} \text{ Incidentes}} feromona_{Arcoi \rightarrow z(Roja)}}{n^{\circ} \text{ Incidentes}} - \frac{\sum_1^{n^{\circ} \text{ Adyacentes}} feromona_{Arcoi \rightarrow j(Verde)}}{n^{\circ} \text{ Adyacentes}}$$

- Modificación comportamiento de la hormiga fiel. En el primer modelo presentado, se contempla una hormiga denominada fiel para seguir el camino de mayor valor de feromona. En este nuevo diseño, también se necesita un tipo de hormiga similar que partiendo del inicio, y siguiendo en diferentes momentos la feromona verde o roja, consiga llegar al punto destino.

Una vez identificadas las modificaciones a realizar para la elaboración del algoritmo, se procede a determinar que partes o funciones son inevitables de modificación.

En la siguiente ilustración, partiendo del algoritmo implementado en el [punto 3.2.2.2.2](#) del presente proyecto, se reseñan aquellas funciones en las que se han introducidos los cambios que se acaban de exponer, de tal manera que el nuevo algoritmo cumpla con las novedades planteadas.

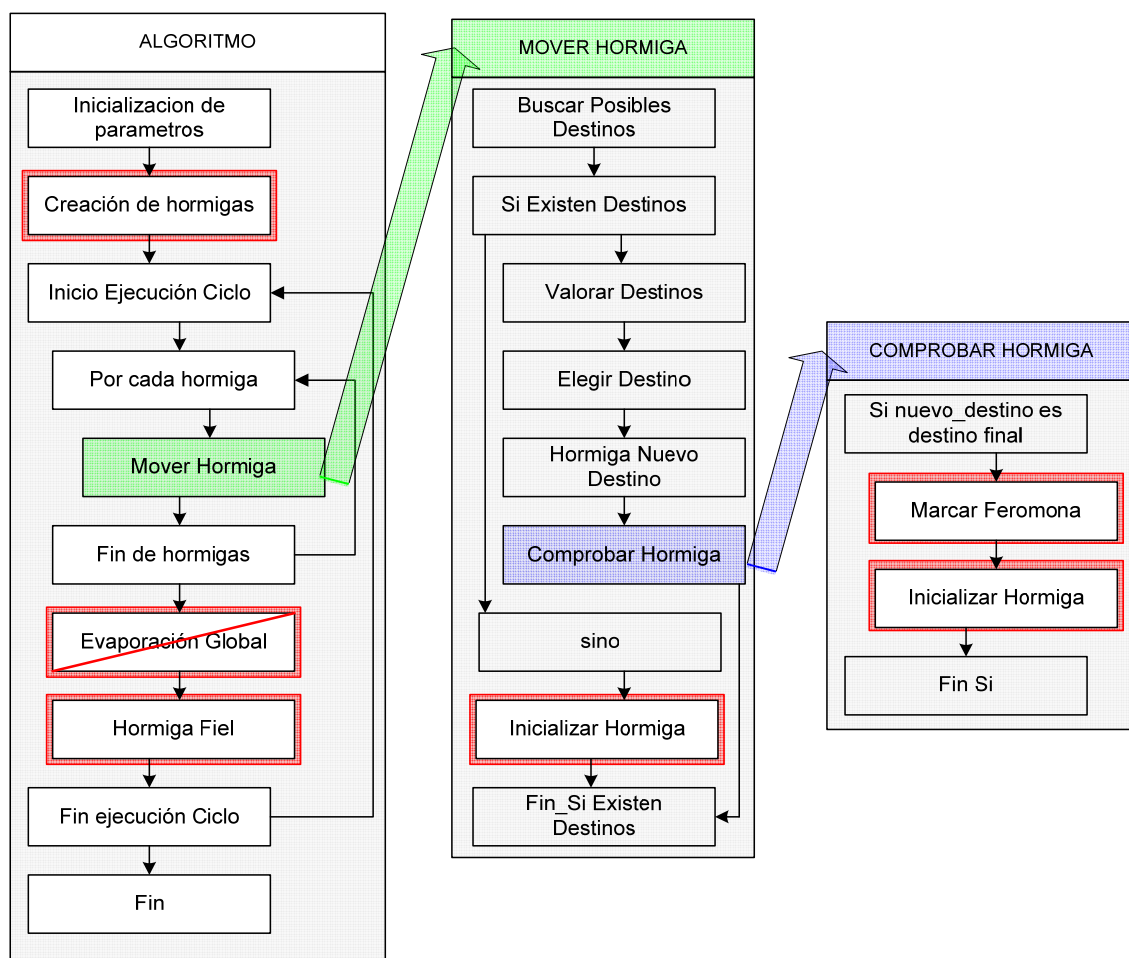


Ilustración 29 – Algoritmo ACO-División del problema.

En un recuadro rojo se indican las funciones del algoritmo donde se han realizado modificaciones correspondientes a los puntos anteriormente mencionados.

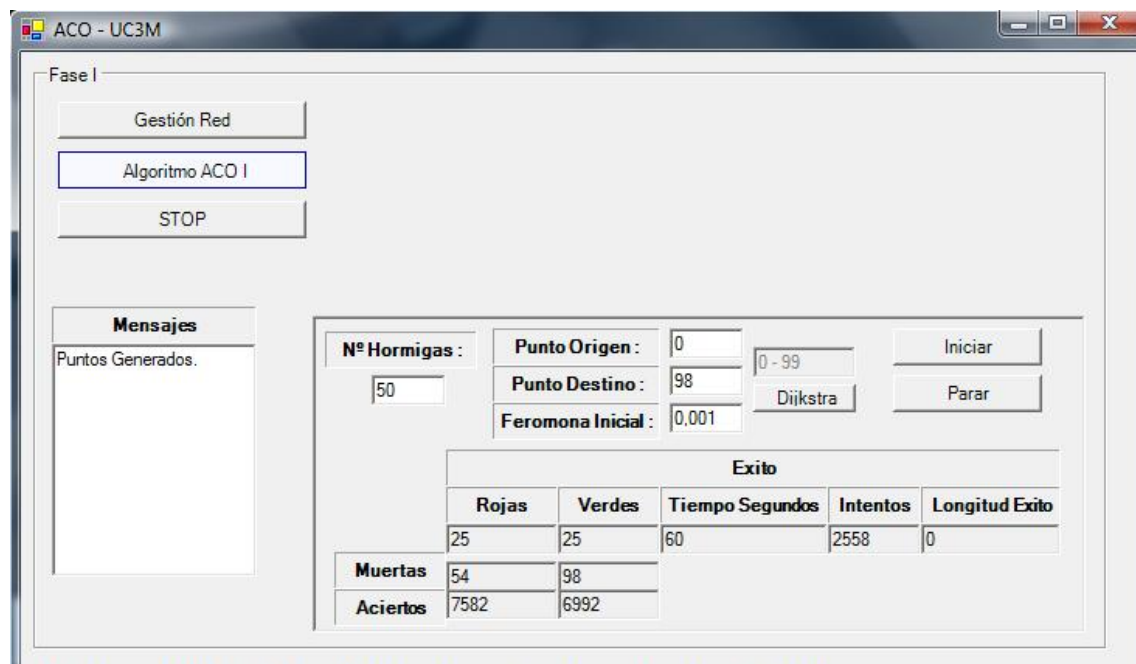
3.3.3 RESULTADOS.

El ejemplo que se va a exponer corresponde al siguiente entorno experimental:

ENTORNO EXPERIMENTAL	
•	Se genera una red de 100 puntos.
•	Cada punto tiene como origen 2 arcos. Cada arco tiene como destino un punto aleatorio de la red.
•	Se crean 50 hormigas dispuestas de forma aleatoria en los puntos de la red. El 50% de las hormigas, 25 en este caso, tienen como misión la búsqueda del origen que se designe y el resto la búsqueda del destino.
•	El valor de feromona inicial es de 0,001.

Tabla 21 – Entorno Experimental. II_R100-2-50.

Los resultados de una ejecución en este entorno se observan en la siguiente imagen.



Exito					
	Rojas	Verdes	Tiempo Segundos	Intentos	Longitud Exito
	25	25	60	2558	0
Muertas	54	98			
Aciertos	7582	6992			

Ilustración 30 – Resultados. Modelo II_R100-2-50.

Se puede ver como los aciertos de los miembros de la colonia, es decir hormigas que consiguen cumplir su misión, son considerables en número, 7582 rojas (búsqueda del origen) y 6992 verdes (búsqueda del destino). El número de intentos que se ha lanzado la hormiga fiel desde el origen hasta el destino han sido 2558 y de ellas ninguna vez ha conseguido tener éxito.

A diferencia del modelo anteriormente expuesto, en este nuevo modelo las hormigas de la colonia no proporcionan ningún resultado, ya que su recorrido se realiza desde puntos aleatorios de la red a los puntos origen o destino según la función que desempeñan en la colonia.

El resultado viene proporcionado por la hormiga fiel, que intenta viajar desde el origen al destino, siguiendo los valores depositados de feromona por las hormigas de la colonia y dependiendo de la aleatoriedad de posicionamiento de los diferentes puntos creados, puede o no conseguir su objetivo.

○ Conclusiones

Los resultados en esta fase fueron nefastos, los éxitos eran completamente aleatorios y en tamaños de red pequeños. Como se observa en el ejemplo presentado en la ilustración anterior, los éxitos de los dos tipos de hormiga son normales, es decir la proporción entre hormigas que alcanzan el éxito de su misión con aquellas que no lo consiguen es normal. El problema reside en que la hormiga fiel no logra su objetivo, unir el origen con el destino.

3.3.4 PROBLEMAS ENCONTRADOS.

El principal problema que se detectó, y originaba que los éxitos fueran aleatorios, era debido a la formación del camino en la zona del origen, es decir, seguimiento de la feromona roja. Si la distancia entre los puntos origen y destino era pequeña, el encontrar un punto verde, es decir perteneciente al destino, era factible, en cambio si la distancia aumentaba, la posibilidad de avanzar hacia la zona verde era exigua, aumentando, por consiguiente, la posibilidad de perderse, con lo que no se conseguía solución o esta estaba bastante alejada del óptimo.

El problema era como salir de los puntos marcados como rojos. Si se realizaba el recorrido siguiendo los puntos con mayor feromona roja, existía mucha probabilidad de realizar una vuelta al origen. Si por el contrario, se buscaba puntos en los que la feromona roja fuese menor, no me aseguraba que fuera en el sentido correcto del destino de la ruta. Al final, para obtener un buen resultado en la decisión, siempre intervenía la cantidad de feromona verde depositada, que es la que acerca al destino. Las hormigas de búsqueda del origen no aportan una mejora directa para la resolución del problema. Esta fue la causa principal por la que se consideró descartar esta primera idea de trabajo.

Las hormigas en el transcurso del tiempo generaban una especie de curvas de nivel, representando distancias al origen y destino. Al generar círculos, la dirección entre origen y destino es una variable que no se puede identificar y en consecuencia no hay garantías de obtener una solución correcta al problema.

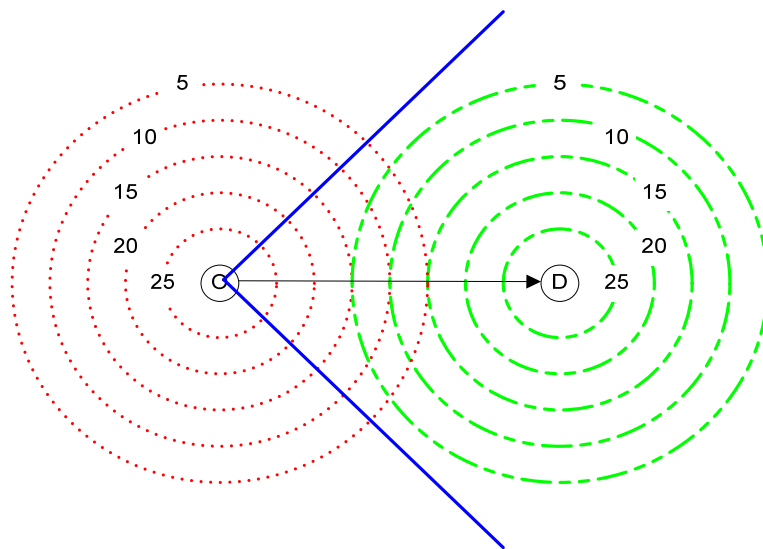


Ilustración 31 – Comportamiento Algoritmo.

Como se observa en la ilustración, la solución sería buscar valores de feromona siempre menores dentro de las posibilidades de desplazamiento que se presenten en un punto de los considerados rojos, o más cercanos al origen. El problema es la dirección que marque la elección de los puntos. Si la dirección elegida no es hacia la zona del punto destino, las probabilidades de éxito son escasas y la pérdida de la hormiga fiel asegurada.

Una posible solución que se intento implementar para subsanar este error es la siguiente:

- La hormiga fiel tenía que partir de aquellos puntos donde el valor de la feromona de ambos grupos fuese similar y de mayor valor, avanzando hacia ambos lados, originando al final una sola ruta que uniese origen y destino. Sería encontrar una zona frontera y realizar una búsqueda en ambos sentidos, hacia puntos de mayor cantidad de feromona depositada, pero esta solución añadía complejidad al algoritmo, cosa que no debería de ocurrir por eficiencia.

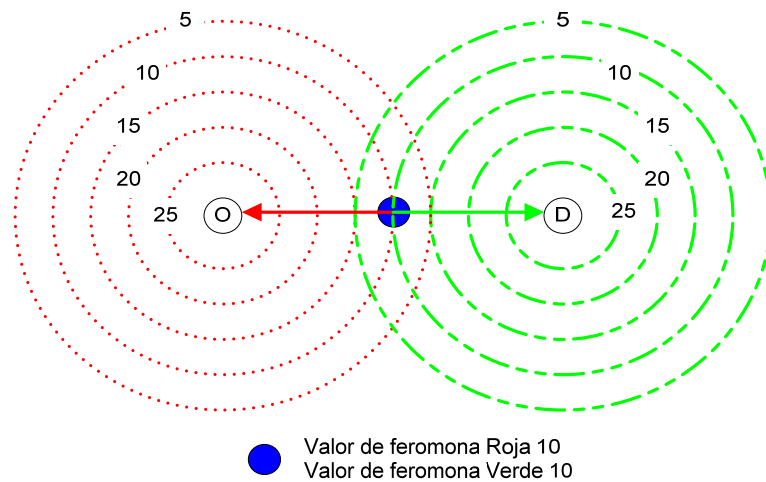


Ilustración 32 – Posible solución al modelo planteado.

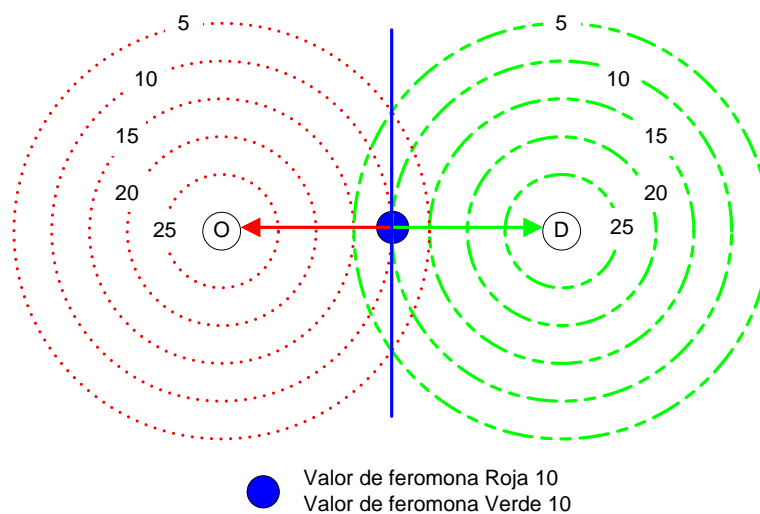
El punto buscado tiene que intentar cumplir las siguientes expresiones.

$$Tipo_Punto_i = feromona_{Roja-i} - feromona_{Verde-i} \approx 0$$

$$Max(\sum_1^n feromona_{Arcoi \rightarrow i(Roja)})$$

$$Max(\sum_1^n feromona_{Arcoi \rightarrow h(Verde)})$$

Las dos feromonas presentan un valor similar, es decir el campo tipo del punto tiene que tener un valor aproximado a 0. Esto restringe el espacio de búsqueda, de una manera teórica, a una línea que separa ambos puntos, como se observa en la siguiente ilustración.

**Ilustración 33 – Posible solución al modelo planteado II.**

Para buscar el mejor valor en la línea, hay que centrarse en buscar el valor máximo de ambas feromonas, aunque simplemente la búsqueda de una sería suficiente, ya que la primera condición obligaría a que los valores de ambas feromonas fuesen iguales.

Aunque la solución aportada puede ser factible, se busca un procedimiento simple, no se podía complicar el algoritmo en búsqueda de soluciones a nuevos problemas que originaban una mayor complejidad, razón por la que se desechó el modelo.

3.4 ORIENTACIÓN DE LA COLONIA DE HORMIGAS EN EL DESCUBRIMIENTO DEL DESTINO.

Un nuevo concepto interesante introducido en el modelo anterior es la distribución aleatoria de las hormigas por la red. Este concepto se intenta mantener en este nuevo diseño.

Se ha observado que cuando la distancia entre el origen y destino es grande, existiendo bastantes puntos intermedios, la búsqueda del punto destino tarda un tiempo considerable. Los siguientes resultados, realizados en el modelo propuesto en el [punto 3.2](#), sirven a modo de ejemplo:

ENTORNO EXPERIMENTAL	
•	Se genera una red de 1000 puntos.
•	Cada punto tiene como origen 20 arcos y como destino de cada arco un punto aleatorio de la red.
•	Se crean 50 hormigas en el origen.
•	El valor de feromona inicial es de 0,01.
•	Evaporación 1%.

Tabla 22 – Entorno experimental. I_R1000-20-50.

Puntos en la red	1000		
	Mejor Resultado (Colonia)		
Nº Arcos por punto	20		
	Ciclo	Tiempo	Distancia
1 ^{er} Éxito	178	5750	986
2 ^o Éxito	181	6032	969
Optimo	179		

Ilustración 34 – Resultados I_R1000-20-50. Comportamiento Algoritmo. 1^{er} Éxito.

Como se puede observar para 1000 puntos se necesitan 178 ciclos para encontrar el destino (primer éxito), con una longitud o peso de 986, es decir muy alejada del óptimo que es 179. Esto verifica que la primera búsqueda es muy costosa. Las hormigas durante su recorrido no detectan ningún tipo de información y fácilmente se pierden en la red. El poder ayudar en esta búsqueda inicial es lo que se intenta con esta nueva versión.

Si se dispone de una serie de hormigas repartidas por toda la red se facilitará el descubrimiento del destino por alguna de ellas. Una vez encontrado, esta marcará con feromona su ruta, con lo que las siguientes hormigas, en la búsqueda que realizan, tendrán más posibilidades de encontrarse con arcos que la guíen hacia el destino.



Se puede decir que el conjunto de hormigas, con situación aleatoria en la red, servirán de guías hacia el destino del resto de individuos de la colonia.

3.4.1 MODIFICACIONES PLANTEADAS.

Partiendo del modelo implementado en el [punto 3.2](#), de este documento, se han considerado una serie de modificaciones con el fin de mejorar los resultados obtenidos hasta el momento. Las modificaciones se formulan en los siguientes puntos:

- Se va a disponer una serie de hormigas situadas de manera aleatoria por toda la red. Con esto se intenta mejorar, en un primer momento, la búsqueda del destino disminuyendo el tiempo que tarda el algoritmo en encontrar la primera solución, y posteriormente, caminos más eficaces para la resolución del problema.
- Por la razón del punto anterior, existirán dos tipos de hormigas. Un conjunto se situará aleatoriamente por la red y el otro partirá del origen como en el modelo original propuesto en el [punto 3.2](#). El valor de reparto será una variable del programa.
- Se añade el concepto de “*hormiga muerta*”. Interesa saber aquellas hormigas que no pueden realizar movimiento y no han conseguido llegar al destino. De igual manera interesa conocer el número de hormigas que han visto cumplido su objetivo “*hormiga éxito*”.
- Se mantiene el concepto de “*hormiga fiel*” del modelo original.
- Se mantiene, del modelo propuesto en el [punto 3.3](#), la eliminación de la evaporación de feromona.
- Se mantiene la aportación de feromona únicamente a posteriori, es decir una vez concluida con éxito la ruta de una hormiga hasta el destino, siendo siempre función del resultado obtenido.

3.4.2 MODIFICACIONES REALIZADAS.

En este punto, se intenta realizar una explicación detallada de como las modificaciones planteadas se plasman en el nuevo algoritmo. Las modificaciones realizadas se pueden agrupar en dos ideas y son las siguientes:

- Inicialización de Parámetros:

Porcentaje en origen: Se introduce una variable que cuantifica la cantidad de hormigas en el origen, como consecuencia también se identifica el número de hormigas del otro grupo, es decir las que tienen una disposición aleatoria por la red.

- Creación de Hormigas:

Se realiza la creación de hormigas, repartidas en los dos tipos existentes, dependiendo de la variable introducida en el programa. Esta variable presenta un valor en tanto por ciento del número total de hormigas que forman la colonia.

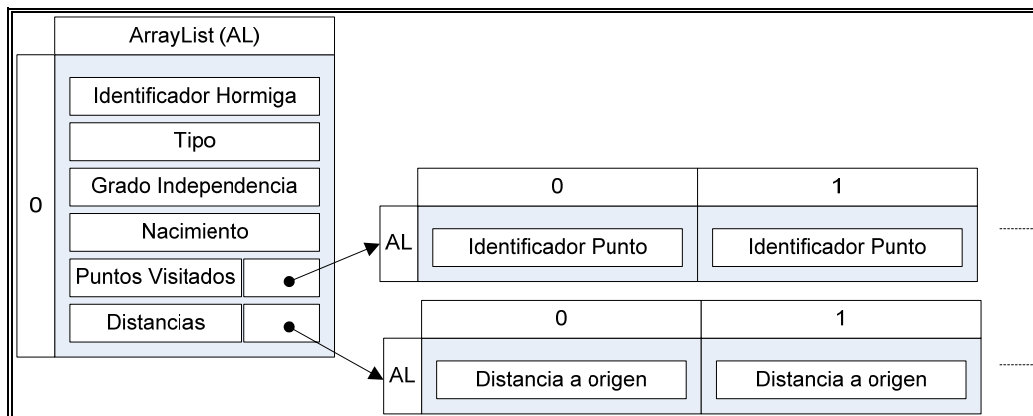


Ilustración 35 – Estructura Hormiga.

Se utiliza el campo "tipo" para identificar la clase de hormiga que se crea. La variable "Porcentaje en origen" introducida por el usuario, marca el número de elementos de cada grupo, creando la hormiga en el punto origen (valor 1) o en un punto aleatorio de la red (valor 0). Esta diferenciación sirve para una vez conseguido el éxito de una hormiga determinada, su nueva creación sea del mismo grupo y en el modo adecuado.

3.4.2.1 MODIFICACIÓN DEL ALGORITMO

El algoritmo del modelo desarrollado quedaría, tomando como base el primer modelo propuesto ([punto 3.2.](#)), como indica la siguiente ilustración, señalando con un recuadro en rojo las funciones donde se realizan las modificaciones consideradas:

Se modifican tres funciones del algoritmo inicial para satisfacer las modificaciones planteadas en el punto anterior. Las tres funciones son las siguientes:

- Inicializar Parámetros. Se crean dos conjuntos de hormigas con diferente función. Es un parámetro de inicialización del algoritmo.
- Creación de hormigas. El nacimiento de las hormigas es diferente según el tipo de hormiga creado y el parámetro del punto anterior.
- Inicializar hormiga. Inicialización cuando una hormiga concluye su trayecto

Como en el punto anterior y como variación del algoritmo tomado como base, se prescinde de la función de evaporación por el consumo computacional que origina su ejecución en este tipo de problemas, como se expuso en el [punto 3.2.4](#) del presente documento.

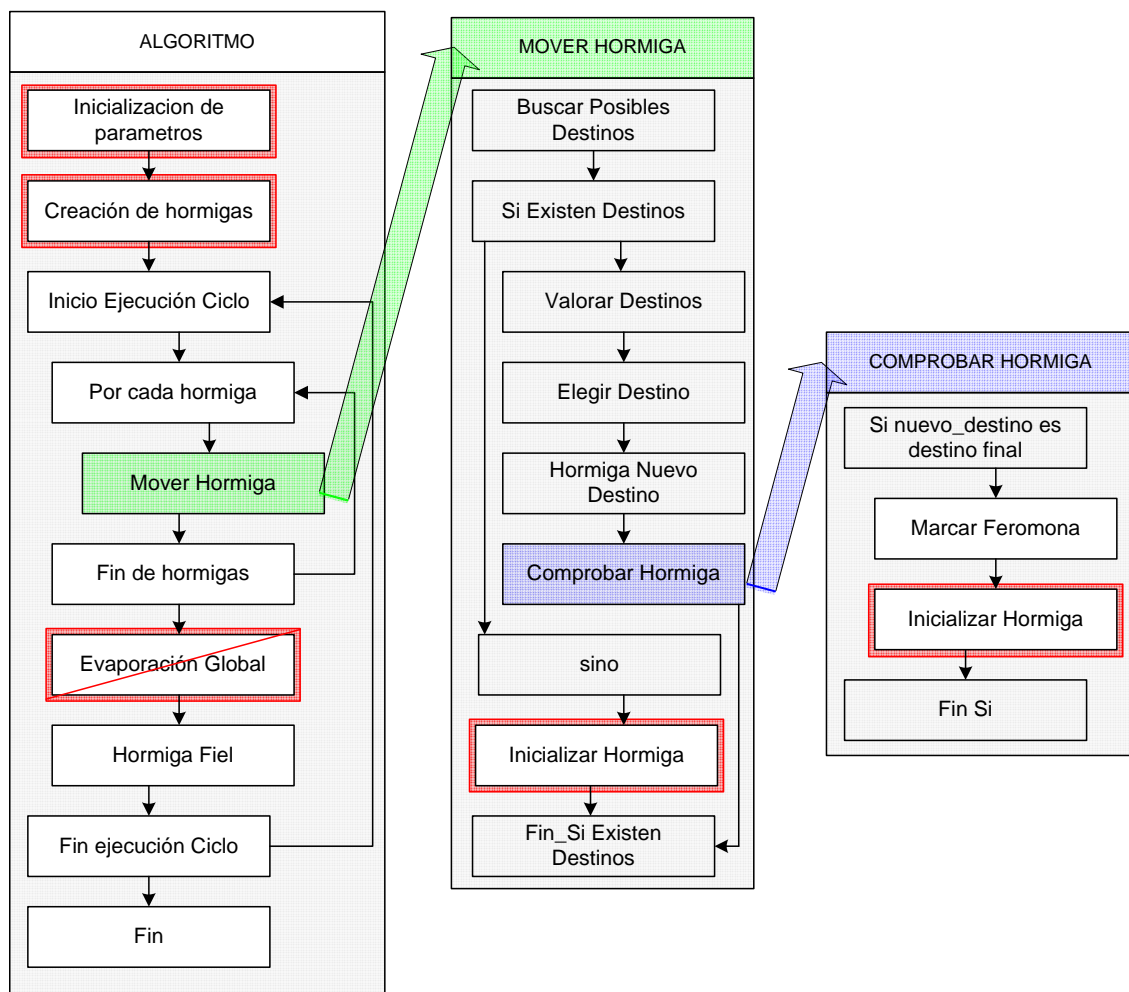


Ilustración 36 – Algoritmo ACO-Orientación de la Colonia de Hormigas.

Son pocas las funciones modificadas respecto al algoritmo tomado como base y se centran en la disposición inicial de las hormigas durante la ejecución del algoritmo.

La descripción sobre las diferentes funciones que forman el algoritmo se describe en el [punto 3.2.2.2.2](#) anteriormente desarrollado.

3.4.3 RESULTADOS.

En esta fase, se ha intentado como objetivo una mejora en eficacia del algoritmo para este tipo de problema en particular. Dentro del objetivo del proyecto figura el estudio del comportamiento del algoritmo cuando la red aumenta en tamaño. El encontrar cuanto antes la primera solución y el mejorar la aproximación al destino, se considera fundamental para optimizar los resultados cuando el problema crece en tamaño.

Como ejemplo de los resultados obtenidos en esta fase y que han servido para la identificación de errores y por consiguiente, originando un conjunto de mejoras que se implantarán y explicarán a posteriori, mostramos la siguiente ilustración pasándola a explicar posteriormente:

• Resultados Modelo. III_R100-2-50.

ENTORNO EXPERIMENTAL	
•	Se genera una red de 100 puntos.
•	Cada punto tiene como origen 2 arcos y como destino de cada arco un punto aleatorio de la red.
•	Se crean 50 hormigas, aproximadamente el 50% en cada grupo.
•	El valor de feromona inicial es de 0,001.

Tabla 23 – Entorno experimental. III_R100-2-50

Puntos en la red		100					
		Hormiga Fiel			Mejor Resultado (Colonia)		
Nº Arcos por punto		2			2		
		Ciclo	Tiempo	Distancia	Ciclo	Tiempo	Distancia
1 ^{er} Éxito		16	281	84	16	281	84
		20	343	95	20	343	83
		37	640	98	37	640	79
		38	656	100	38	656	76
		56	921	91	55	921	75
		59	984	85	59	984	74
Último Éxito		106	1718	94	106	1718	71
Óptimo		71					

Ilustración 37 – Resultados. III_R100-2-50.

Este ejemplo permite identificar el avance conseguido de versiones anteriores. A la vez descubre carencias y errores que presenta el algoritmo objeto de este punto.

Se observa que el programa es capaz de alcanzar el óptimo en menos de dos segundos de ejecución, un avance respecto a la versión inicial. En el ciclo 106 al menos una hormiga ha sido capaz de obtener la mejor distancia. Se observa la evolución de las distancias, desde un inicio de 84, como evoluciona hasta conseguir el óptimo.

La ilustración descubre más cosas, entre ellas vemos que la hormiga fiel, que sigue los rastros de mayor feromona de los caminos que encuentra, hay veces que empeora sus resultados. Esto quiere decir que el camino más probable no es el mejor hasta el momento descubierto.

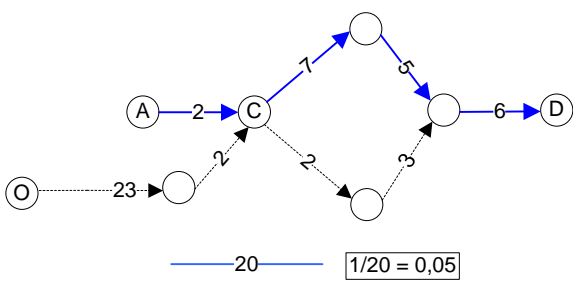
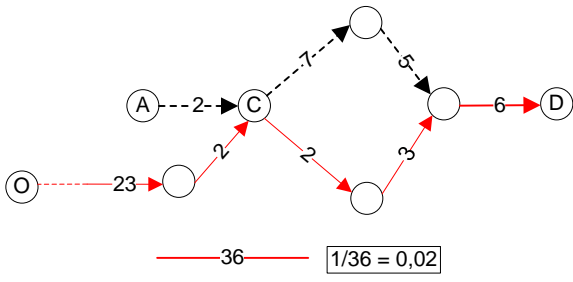
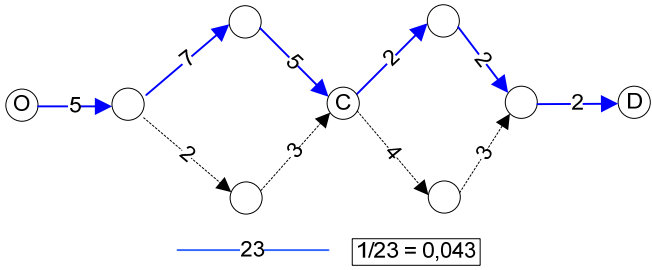
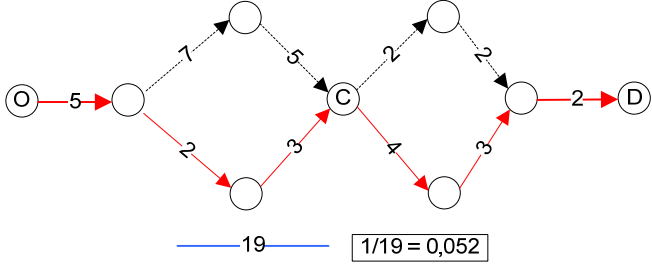
A pesar de comprobar mejoras en los resultados, durante la fase de desarrollo y verificación, se han observado una serie de errores que conviene subsanar y que a continuación se detallan de una manera más explícita y que se subsanarán en el siguiente modelo implementado.

3.4.4 PROBLEMAS ENCONTRADOS.

• Depósito de Feromona:

Es un problema bastante serio que se ha detectado. Cuando tenemos hormigas que parten de diferentes puntos, la deposición de la feromona en función de la distancia

recorrida hasta el destino es un error. El siguiente ejemplo lo muestra de manera gráfica para una mejor comprensión.

ERROR DEPOSICIÓN DE FEROMONA	
ruta realizada	DESCRIPCIÓN
	<p>Se inicializa una hormiga en una posición aleatoria de la red (A), realizando el recorrido marcado en línea continua azul hasta llegar al destino (D). El recorrido efectuado representa un peso de 20, con lo que el aporte de feromona en los arcos es de una cantidad de 0.05.</p>
	<p>Otra hormiga, en este caso partiendo del origen (O) realiza el recorrido hasta llegar al punto (C), siguiendo a partir de este punto una ruta mejor que la anterior hasta llegar al destino (D), en total un peso de 36. Como se observa, el camino de C-D en este caso es mejor, en cambio el aporte de feromona es menor que en el caso anterior.</p>
<p>Este error surge en diferentes situaciones, no sólo cuando hay hormigas aleatorias, también si se parte del origen. Es un error de este conjunto de problemas donde el número de arcos y pasos intermedios es variable.</p>	
	<p>En este caso la hormiga parte del origen y desarrolla una ruta total con un peso de 23. Pero hay que indicar que desde el punto C al destino el peso es de 6.</p>
	<p>En este otro caso la hormiga realiza una ruta más corta entre el origen y destino, en total 19. Pero la distancia entre C y D en este caso es mayor. El aporte de feromona en la ruta es mayor que en el ejemplo anterior, en cambio parte de la ruta es menos efectiva.</p>

ERROR DEPOSICIÓN DE FEROMONA	
RUTA REALIZADA	DESCRIPCIÓN
Es un error del modelo, el aporte de feromona en función del resultado obtenido en la ruta puede dar lugar a error. Teniendo en cuenta que el objetivo de nuestro estudio es comprobar el comportamiento de estos algoritmos en tamaños de red grandes, y siendo este un problema que aumenta en probabilidad con el tamaño de la red, esta inexactitud anuncia un éxito relativo o la búsqueda de nuevas soluciones. Incluso la hormiga fiel no sería capaz de detectarlo y presentaría un resultado erróneo como ha ocurrido en los resultados mostrados en este punto.	

Tabla 24 – Error deposición de feromona. Problemas encontrados.

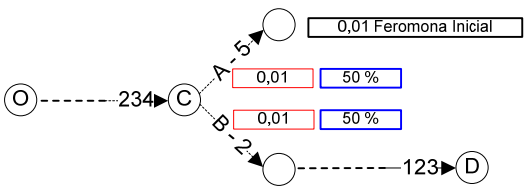
Estos son dos ejemplos de errores que se producen en la deposición de feromona para el modelo en estudio que pueden conducir a la determinación de soluciones ineficaces.

- Feromona Inicial:

El depósito de feromona inicial también representa un problema a resolver en rutas con longitudes variables. No se puede establecer una cantidad inicial para todo tipo de instancias del problema.

Si tengo una red donde puedo preguntar rutas mínimas sobre diferentes ciudades, en algunos casos la distancia entre puntos puede ser elevada y en otros caso cortas. Esto origina que los depósitos de feromona, que se realizan en función de la distancia entre origen y destino, pueden variar de orden en la misma red.

El depósito de feromona inicial tiene que depender de la distancia a recorrer en la instancia del problema para que su significado sea coherente.

ERROR DEPÓSITO DE FEROMONA INICIAL	
RUTA SOLICITADA	DESCRIPCIÓN
	<p>Partiendo de que la red es grande, en un momento dado me interesa conocer la distancia mínima entre dos puntos (O - D). Desconozco la distancia que los separa, y la cantidad de feromona inicial que presentan mis caminos es de 0,01. En un primer momento, la primera hormiga que llega al punto C, se encuentra que tiene la misma probabilidad (50 %) de elegir el camino A o el B. Una vez llega al destino final D, concluye su ruta con una distancia de 359 entre O y D. Por lo tanto realiza una puesta de feromona en el camino recorrido en función de esta distancia, es decir $1/359$.</p>

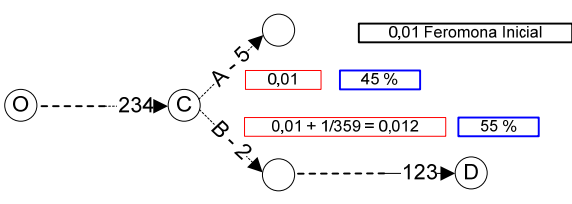
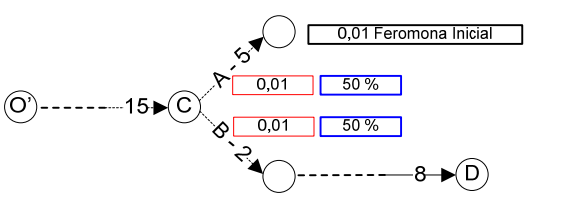
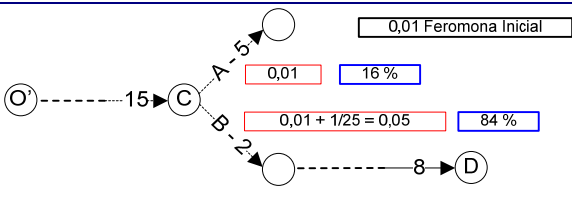
ERROR DEPÓSITO DE FEROMONA INICIAL	
RUTA SOLICITADA	DESCRIPCIÓN
	<p>Posteriormente, otra hormiga llega al punto C, teniendo ahora una de las rutas un aporte extra de feromona debido al éxito de la hormiga anterior. La probabilidad de elección varía, ahora le corresponde un 45 % para el arco A y un 55% para el arco B.</p>
<p>Este es un caso posible solicitado a nuestro software. Si nosotros ahora queremos realizar otra búsqueda entre otros dos puntos, del que también desconocemos su distancia ocurriría lo siguiente.</p>	
	<p>La primera hormiga cuando llega al punto C, como en el caso anterior, la probabilidad de elegir entre dos rutas (A, B) es del 50% cada una. Si elige la B y consigue llegar al destino, realizará un aporte de feromona en relación con la distancia de éxito, es decir 1/25 (0,04)</p>
	<p>La siguiente hormiga en llegar al punto C, en este caso tendrá una probabilidad para elegir ruta mucho más sesgada que en el ejemplo anterior, ya que sería de un 16% para la ruta A y un 84% para la ruta B.</p>
<p>Este ejemplo es otro error que presenta el modelo. La feromona inicial tiene que tener una relación directa con cada instancia del problema, o más concretamente con la distancia de cálculo, que es nuestra incógnita. Se puede realizar una estimación, como en algunos modelos documentados se ha insinuado, pero además el tipo de problema objeto de estudio, presenta la desventaja de la posibilidad de realizar consultas sobre diferentes longitudes en la misma red.</p> <p>Esto nos lleva a preguntarnos, ¿Cada consulta realizará una estimación de la distancia?. Una vez estimada, ¿se procederá a realizar la variación de la feromona inicial en todos los arcos?. La solución puede presentar una complejidad considerable, efecto que no admiten este tipo de algoritmos.</p>	

Tabla 25 – Error feromona inicial. Problemas encontrados.

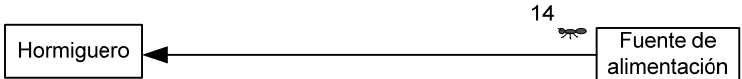
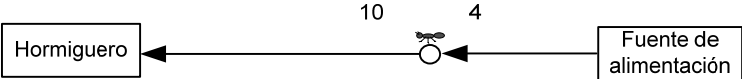
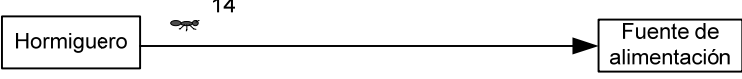
El valor de la feromona inicial influye decisivamente en las probabilidades de elección de rutas de las hormigas en el transcurso de resolución de una instancia del problema. Cada consulta tiene que tener un valor de feromona inicial representativo y relacionado con la instancia del problema.

3.5 ESTUDIO SOBRE LOS DEPÓSITOS DE FEROMANA EN BASE A UNA REFERENCIA ÚNICA.

Este nuevo modelo trata de buscar una solución al problema que se detecto en la fase anterior relacionado con el aporte de feromona, tanto el inicial, como la deposición una vez concluida con éxito la ruta. La solución tiene que ser simple y que no complique el algoritmo en demasía para que este mantenga sus prestaciones.

Basándonos en unos estudios realizados sobre la especie de hormiga *Cataglyphis fortis* realizados por Harald Wolf, neurobiólogo de la Universidad de Ulm, se intento solucionar el problema que se presenta con el depósito de feromonas. Se necesita tener una referencia única en cada punto para que los valores de feromona sean representativos de la deseabilidad de la ruta a elegir.

Para ello, se simula en cada punto el podómetro que permite a esta especie de hormiga conocer la distancia entre el hormiguero y la fuente de alimentación. El conocimiento de la distancia al destino en nuestro medio consistirá en una determinada cantidad de feromona, esto permite tener referencias únicas, desde cada punto al destino en todo momento, como se detallará más detenidamente en el punto siguiente. Una explicación grafica puede ayudar a su comprensión.

SIMULACIÓN DEL COMPORTAMIENTO DE LA HORMIGA CATAGLYPHIS FORTIS	
1	 <p>La hormiga <i>Cataglyphis fortis</i> en su regreso desde la fuente de alimentación conoce la distancia que le separa del hormiguero.</p>
2	 <p>Según recorre el trayecto en cada momento sabe la distancia que le queda por recorrer.</p>
3	 <p>Cuando la hormiga regresa desde el hormiguero a la fuente de alimentación, sabe la distancia que le separa y la dirección a seguir para conseguir el alimento</p>
INTRODUCCIÓN DEL COMPORTAMIENTO DE LA HORMIGA CATAGLYPHIS FORTIS EN NUESTRO MODELO	
<p>En el modelo hasta el momento implementado, toda la información entre los miembros de la colonia se transmite a través del medio por medio de aportes de feromona. Este medio de comunicación será el utilizado en el nuevo modelo, donde la feromona de cada punto informará de la distancia que separa del objetivo o meta. La marca de feromona en los arcos de regreso por parte de las hormigas de la colonia informará tanto del trayecto seguido, como de la distancia recorrida para su utilización por el resto de miembros de la colonia.</p>	

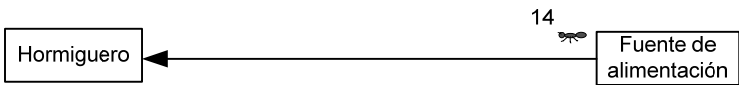
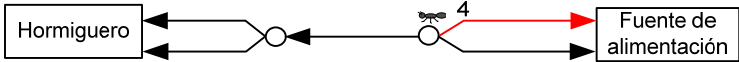


SIMULACIÓN DEL COMPORTAMIENTO DE LA HORMIGA CATAGLYPHIS FORTIS	
1	 <p>Una hormiga de la colonia cuando llega a su destino conoce tanto la distancia recorrida como la ruta seguida, toda esta información recogida en su lista de ciudades visitadas (lista tabú).</p>
2	 <p>En su regreso, el depósito de feromona que realiza a posteriori, marca cada arco de su recorrido con una cantidad de feromona relacionada con la distancia recorrida.</p>
3	 <p>El depósito de feromona lo realiza en cada arco perteneciente a su recorrido.</p>
4	 <p>Una vez concluida la ruta toda la información queda reflejada para la utilización por el resto de miembros de la colonia.</p>

Tabla 26 – Simulación del comportamiento de la hormiga Cataglyphis fortis.

La representación de este comportamiento en el algoritmo se verá explicada más detenidamente es los puntos siguientes.

3.5.1 MODIFICACIONES PLANTEADAS.

El nuevo algoritmo conllevará algunos cambios, en especial en el tratamiento de la feromona, grave problema detectado en el punto anterior, y que se intentara resolver con el aporte de nuevo conocimiento, en este caso la posibilidad de presentar una referencia única por punto para cada instancia del problema.

- **Depósito de Feromona**

Van a existir dos tipos de feromona. La denominación que se ha decidido utilizar es "*Feromona Inicial*", que origina una probabilidad aunque mínima de exploración de diferentes rutas y "*Feromona Depositada*", que está directamente relacionada con la distancia al destino de la ruta que transcurre por dicho tramo. Es una simulación del podómetro que presenta la hormiga inspiradora del modelo.

En un inicio, todos los arcos tienen que tener una cantidad igual de feromona para realizar la exploración de manera equitativa. Siempre se han utilizado en los algoritmos basados en colonias de hormigas las dos feromonas, pero no de forma independiente. Como se ha indicado en los problemas detectados en la fase anterior, el valor de feromona inicial tiene que tener una relación con los valores de feromona depositada para que pueda valorarse según las situaciones o instancias del problema.

- Valores iniciales de las feromonas utilizadas.

La feromona inicial (FI), puede tener en un comienzo un valor cualquiera. En nuestro caso se ha elegido un valor de uno. En cuanto a feromona depositada (FD) el valor tiene que ser cero, ya que no ha sido transitado por ninguna hormiga. Un ejemplo de un punto cualquiera de la red se puede observar en la siguiente ilustración.

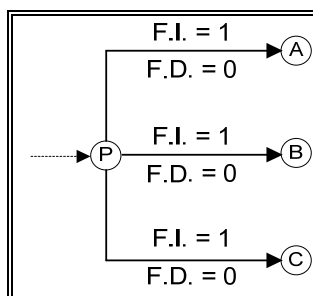


Ilustración 38 – Valores iniciales de feromona.

Estos serían los valores de los diferentes arcos que conforman la red en el estado inicial. Como se observa, los tres arcos que parten del punto P presentan unos valores idénticos de feromona, originando por lo tanto la misma probabilidad de elegir cualquiera de las tres rutas.

- Evolución de los valores de feromona:

Los valores de feromona tienen que tener una referencia única y la solución propuesta toma como base el comportamiento de la hormiga *Cataglyphis fortis*. Nuestra referencia va a ser la mejor distancia conseguida desde el punto en cuestión al destino. La feromona depositada en el arco estará relacionada con la distancia recorrida por la hormiga. Tanto la feromona inicial, como la feromona depositada, tendrán relación directa con la distancia al destino de la mejor hormiga que haya recorrido esa ruta.

- **Hormiga fiel**

Se mantiene el concepto de hormiga fiel siguiendo siempre la ruta donde se detecta mayor feromona. Pero en este caso, significa siempre la búsqueda de la mejor solución, lo cual antes no ocurría, ya que los arcos con un mayor aporte de feromona no siempre eran aquellos que representaban la mejor solución descubierta. Ahora esto se soluciona siguiendo siempre rutas óptimas descubiertas e incluso mejorándolas como se comprobará más adelante.

Estas variaciones planteadas llevan a la realización de unos cambios que a continuación se describen más detenidamente.

3.5.2 MODIFICACIONES REALIZADAS.

Para introducir las novedades anteriormente mencionadas se ha necesitado modificar la estructura de datos del programa, al igual que algunos pasos del algoritmo. A continuación se realiza una exposición de como se han materializado estas modificaciones de una manera más detallada.

3.5.2.1 MODIFICACIÓN DE LA ESTRUCTURA DE DATOS.

Se necesita gestionar nuevos datos para la realización de los cambios planteados. Ha variado el planteamiento, originando una modificación en el intercambio de información entre las diferentes hormigas de la colonia, para ello se necesita modificar tanto la estructura del elemento arco, como de punto. A continuación se detallan estas alteraciones.

- Arcos. Se añade un campo para poder representar la feromona. Ahora existen dos tipos que son necesarios diferenciar, la feromona inicial y la feromona depositada. La nueva estructura de arco queda de la siguiente manera:

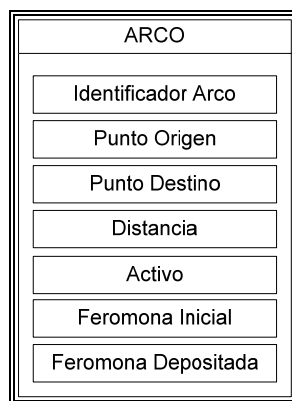


Ilustración 39 – Estructura de datos del elemento Arco.

- Puntos. Para mayor agilidad en la búsqueda de soluciones existentes y tomando como base el comportamiento de la hormiga Cataglyphis fortis, se añade un campo en la estructura de representación de un nodo, indicando siempre la mejor distancia descubierta hasta el destino desde el punto en cuestión.

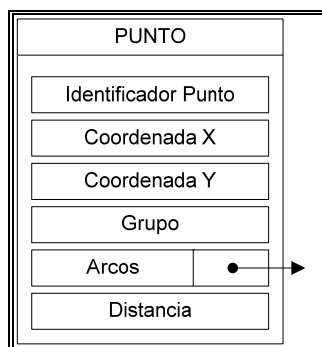


Ilustración 40 – Estructura de datos del elemento Punto.

A la vez que se realiza el depósito de feromona en los arcos del punto se actualiza este valor (distancia), utilizado por la hormiga fiel en su recorrido hacia el destino.

3.5.2.2 MODIFICACIÓN DEL ALGORITMO.

Se ha intentado que los cambios realizados sean lo menos invasores y se respete al máximo lo existente. Partiendo del algoritmo diseñado en el modelo anterior ([Punto 3.4](#)), se han identificado aquellas funciones donde se realizarán los cambios para satisfacer las mejoras planteadas.

Se han señalado con un doble cuadrado en rojo, los lugares del algoritmo donde se van a introducir los cambios en la versión a desarrollar en este punto:

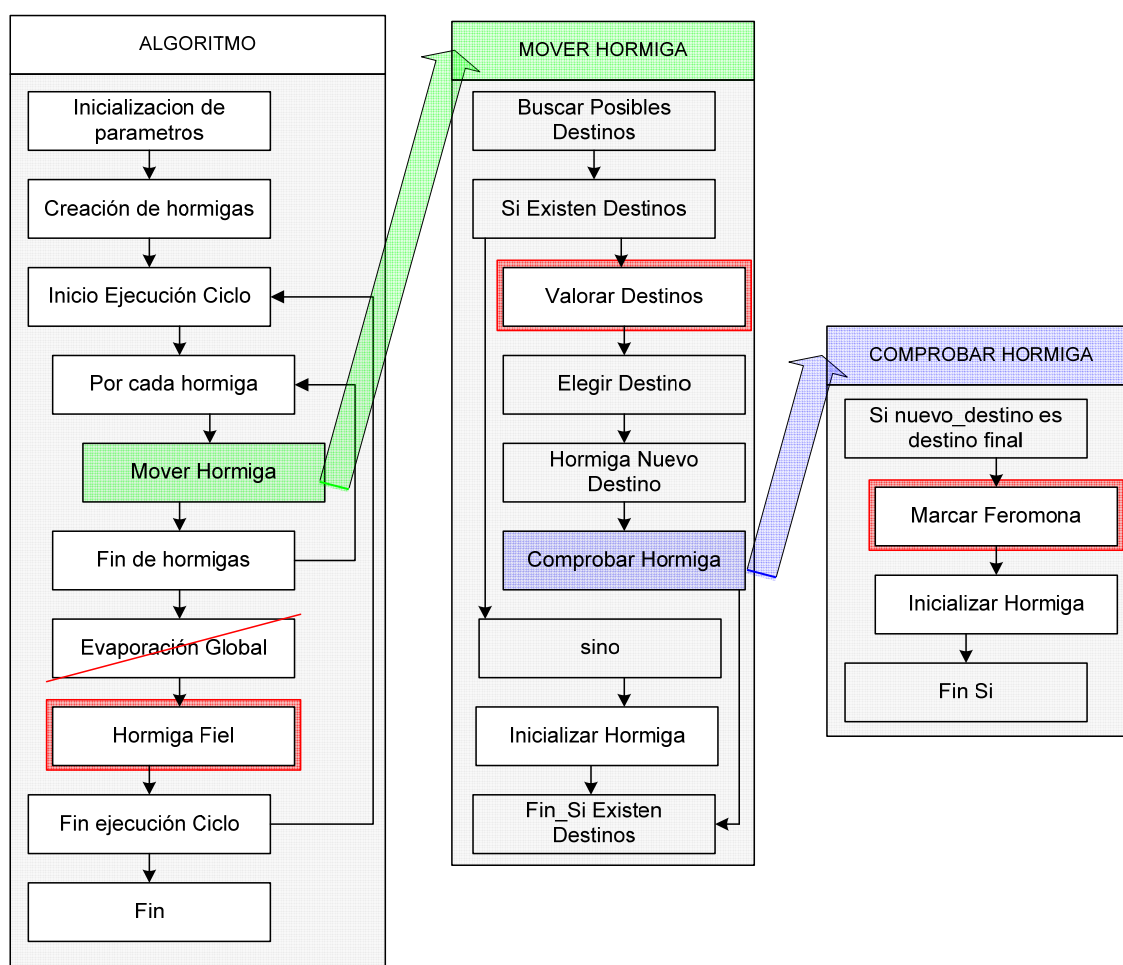


Ilustración 41 – Algoritmo ACO-Referencia única.

Las funciones identificadas en la ilustración anterior están descritas en el [Punto 3.2.2.2.2](#) del presente documento. A continuación se describen aquellas funciones que requieren modificación en la presente versión.

- Valorar Destinos.

La valoración, desde un punto a sus posibles destinos, estará relacionada directamente con la feromona depositada en los arcos correspondientes. La feromona total engloba tanto a la feromona inicial (FI), como la feromona depositada (FD), siendo la primera (FI) función de la menor distancia al destino desde el punto a tratar y la segunda (FD) función de la distancia al destino por el arco objeto de valoración.

$$p_{rs}^k = \begin{cases} \frac{[r_{FI_{rs}}] + [r_{FD_{rs}}]}{\sum_{u \in N_r^k} ([r_{FI_{ru}}] + [r_{FD_{ru}}])}, & \text{si } s \in N_k(r) \\ 0, & \text{No posibles} \end{cases}$$

- Marcar Feromona

El depósito de feromona va a ser único y no acumulativo, hecho que hasta el momento no se estaba realizando. La hormiga realiza una deposición a posteriori en función de la distancia que le separa del punto destino.

La actualización de los dos tipos de feromona se realiza al unísono. La hormiga una vez finalizado su recorrido tiene conocimiento de la ruta seguida y de las distancias entre los diferentes puntos que forman el recorrido con el destino. Además, en cada punto se encuentra almacenada la distancia mínima, hasta el momento descubierta, al destino. Este valor siempre corresponde con el inverso de la feromona inicial de los arcos del punto, pero el valor se almacena en el punto para ganar en eficacia. Con estas premisas, la actualización sigue el siguiente pseudocódigo:


```

MARCAR_FEROMONA (CListaPuntos ListaPuntos, CHormiga Hormiga)
    Distancia_destino = hormiga.Ultimadistancia
    //Recorro todos los puntos por donde se ha desplazado la hormiga
    menos el destino
    For i = 1 to hormiga.NPuntos - 1
        Punto = listaPuntos.punto(hormiga.punto(i))
        Arcos = Punto.listaArcos
        Si punto.distancia > (distancia_destino -
        hormiga.distancia(i)
            Punto.distancia = distancia_destino -
            hormiga.distancia(i)
            For j = 0 to numArcos
                Arco = arcos(j)
                Si arco.destino = hormiga.puntos(i+1)
                    Arco.FD = 1/(distancia_destino-hormiga.distancia(i))
                Fin_si
                Arco.FI = 1/(distancia_destino-hormiga.distancia(i))
            Next for
        Sino
            For j = 0 to numArcos
                Arco = arcos(j)
                Si arco.destino = hormiga.puntos(i+1)and
                arco.FD < 1/(distancia_destino-hormiga.distancia(i))
                    Arco.FD = 1/(distancia_destino-hormiga.distancia(i))
                Fin_si
            Next for
        Fin_si
    Next for
Fin_Marcar_Feromona

```

Ilustración 42 – Algoritmo de deposición de feromona.

Una representación más formal de los valores de feromona, tanto la inicial como la depositada sería la siguiente:

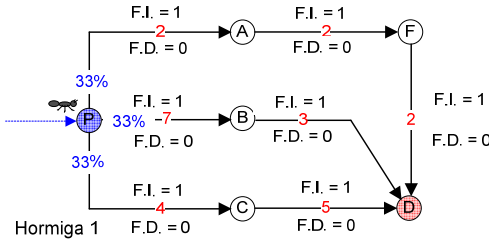
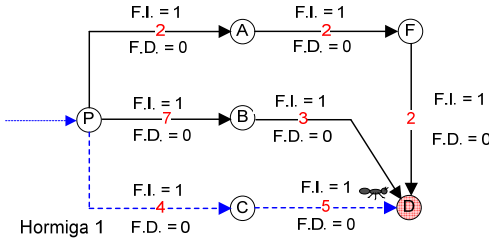
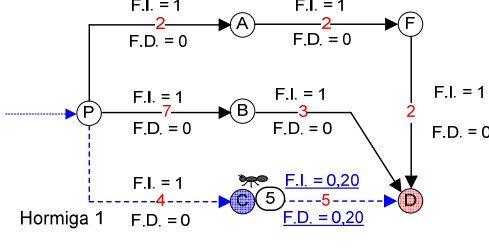
- La feromona inicial sería la misma para todos los arcos que parten del punto i , y sería el máximo valor de feromona que una hormiga haya depositado en cualquier arco que tiene como origen el punto i .

$$FI_{\text{Arcos-Punto}_i} = \text{Max} \left(\frac{1}{\text{Distancia}_{\text{Destino-Punto}_i}} \right)$$

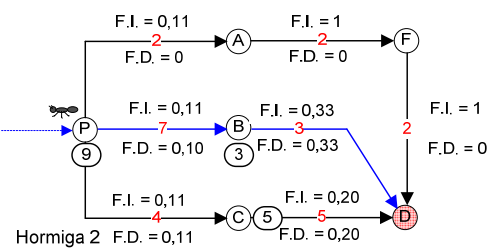
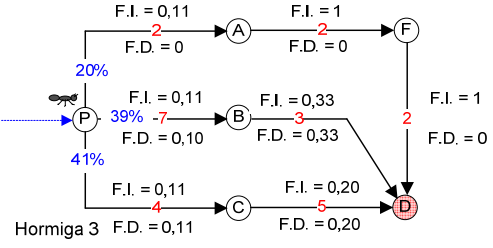
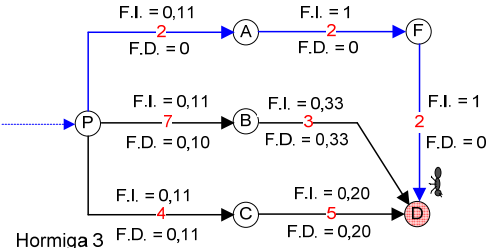
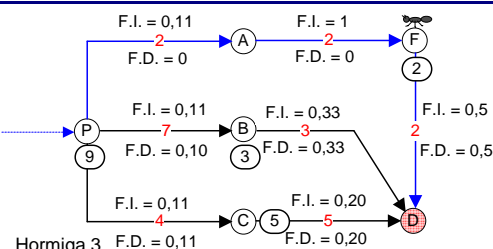
- En relación con la feromona depositada el valor es similar. Este es individual de cada arco y depende de la menor distancia al destino siguiendo el arco en cuestión.

$$FD_{\text{Arco}_i} = \text{Max} \left(\frac{1}{\text{Distancia}_{\text{Destino} \rightarrow \text{Arco}_i}} \right)$$

Tanto la valoración de los destinos, como el depósito de feromona se observa de una manera más ilustrativa siguiendo el estado de un punto determinado durante un periodo de tiempo. A continuación se hace esta exposición que ayudará a la comprensibilidad de lo expuesto anteriormente.

EVOLUCIÓN DE LA FEROMONA EN EL TIEMPO		
	RUTA REALIZADA	DESCRIPCIÓN
1	 <p>Hormiga 1</p>	En un primer momento la hormiga llega al punto P en busca del destino D, teniendo tres posibles rutas para llegar. Al inicio, los arcos presentan una cantidad de feromona inicial (FI) igual, con lo que la probabilidad de elegir cualquier alternativa es la misma
2	 <p>Hormiga 1</p>	La hormiga elige aleatoriamente la ruta P-C-D, para llegar al destino.
3	 <p>Hormiga 1</p>	Una vez conseguido el destino, el algoritmo realiza la deposición de feromona en cada arco recorrido en función de la distancia desde el origen al destino. Así, en el punto (C), realiza una variación en la feromona depositada (FD) en función de la distancia al destino, entre C-D = 5, el valor es por lo tanto $1/5 = 0,20$. Como el valor de feromona inicial (FI) no ha sido modificado, todavía tiene un valor de 1, también realiza una puesta de 0,20 en el campo de la feromona inicial. Además, deja marcado el punto con la distancia al destino, es este caso 5, que siempre corresponde con el inverso de la feromona inicial (FI).

EVOLUCIÓN DE LA FEROMONA EN EL TIEMPO		
	ruta REALIZADA	DESCRIPCIÓN
4	<p>Hormiga 1</p>	<p>La hormiga sigue avanzando hasta el origen y el siguiente punto de paso es P, registrando la distancia al destino, 9 en este caso, y realizando la deposición de feromona en el arco correspondiente. En cuanto a la feromona depositada (FD) tiene un valor de, $1/9=0,11$. En cuanto a la feromona inicial (FI), que al igual que en el caso anterior no ha sido modificada, el valor con el que se actualiza es el mismo, 0,11. Se actualizan todos los arcos que parten de este punto, informando de la distancia más corta detectada hasta el momento desde este punto.</p>
<p>Con esto finaliza el transito de la primera hormiga por el punto P. El siguiente paso es la observación del comportamiento de una segunda hormiga en busca del destino que pasa por dicho punto. En el punto P, los nuevos valores de probabilidad de elección de rutas han cambiado como consecuencia de la variación de los valores de feromona, siendo los siguientes:</p> <div style="display: flex; justify-content: space-around;"> <div>$P-A = 0,11 / 0,44 = 0,25$</div> <div>$P-B = 0,11 / 0,44 = 0,25$</div> <div>$P-C = 0,22 / 0,44 = 0,50$</div> </div>		
5	<p>Hormiga 2</p>	<p>Llega la segunda hormiga al punto P, pero esta vez no existe igual valor de feromona en las opciones que tiene. Existe una mayor probabilidad de elección de la ruta P-C, pero también existe cierta probabilidad de exploración de las otras rutas.</p>
6	<p>Hormiga 2</p>	<p>Elige la ruta P-B-D de manera aleatoria, dentro de la probabilidad existente en cada arco.</p>
7	<p>Hormiga 2</p>	<p>En su camino de vuelta, marca con feromona el arco que une B-D, en base a su distancia al destino, en este caso $1/3=0,33$, tanto en la feromona inicial como la depositada al ser la primera puesta del arco. El punto B lo marca con una distancia de 3.</p>

EVOLUCIÓN DE LA FEROMONA EN EL TIEMPO		
	ruta REALIZADA	DESCRIPCIÓN
8	 <p>Hormiga 2</p>	<p>Llega al punto P con una distancia de 10. Al ser mayor que la existente (9) no realiza ninguna acción. Esto quiere decir que no modifica la feromona inicial en los arcos del punto, pero en cambio si la depositada, donde realiza el aporte correspondiente, 0,10, un valor mayor que el existente, en este momento de 0.</p>
<p>La segunda hormiga que ha pasado por el punto P ha finalizado su ciclo. Ahora el punto presenta dos arcos recorridos y uno todavía no explorado. Vamos a observar el comportamiento de una tercera hormiga en su ruta hacia el destino. Los nuevos valores de probabilidades de elección de ruta en el punto P son los siguientes:</p> <div style="border: 1px solid black; padding: 5px; margin: 5px auto; width: fit-content;"> $P-A = 0,11 / 0,54 = 0,204$ </div> <div style="border: 1px solid black; padding: 5px; margin: 5px auto; width: fit-content;"> $P-B = 0,21 / 0,54 = 0,389$ </div> <div style="border: 1px solid black; padding: 5px; margin: 5px auto; width: fit-content;"> $P-C = 0,22 / 0,54 = 0,407$ </div>		
9	 <p>Hormiga 3</p>	<p>La tercera hormiga llega al punto P y tiene que decidir porque arco continuar su camino. Dos de ellos tienen una probabilidad de elección similar de aproximadamente un 40%, el tercer arco presenta una probabilidad del 20%, es el que todavía no ha sido explorado.</p>
10	 <p>Hormiga 3</p>	<p>La hormiga ha elegido de manera aleatoria, según las probabilidades de cada arco, el recorrido P-A-F-D para llegar a su destino.</p>
11	 <p>Hormiga 3</p>	<p>En su camino de vuelta realiza la deposición de la feromona. En este caso corresponde al arco F-D con una distancia de 2, por lo que realiza un aporte de feromona de $1/2=0,5$, tanto en la depositada como la inicial. Marca el punto con una distancia de 2 al destino.</p>

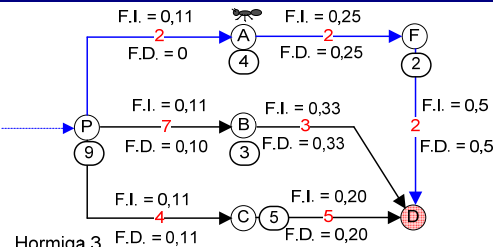
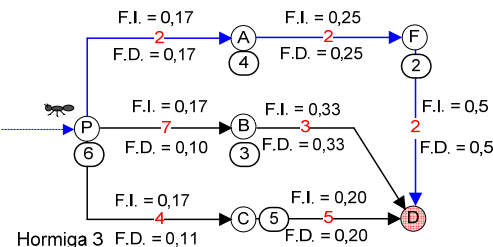
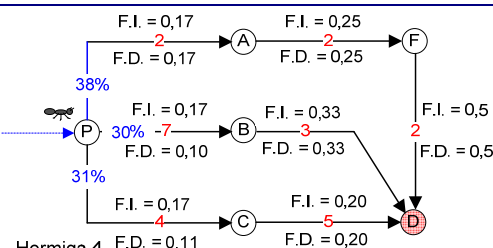
EVOLUCIÓN DE LA FEROMONA EN EL TIEMPO		
	RUTA REALIZADA	DESCRIPCIÓN
12		En su avance hacia el origen la hormiga llega al punto A, en el que realiza una deposición de feromona de $1/4=0,25$ marcando el punto con un valor de 4.
13		La hormiga llega de regreso al punto P, con una distancia de 6, menor que la que hasta el momento presenta el punto, que es de 9. Por esta razón actualiza el punto y también la feromona inicial de todos los arcos que parten del punto, el valor es el inverso de la nueva distancia, $1/6=0,17$. También actualiza la F.D. en el arco correspondiente al nuevo valor.
<p>Los nuevos valores de feromona depositados hacen variar la probabilidades de elección de rutas en el punto P. los nuevos valores son los siguientes.</p> <div style="display: flex; justify-content: center; gap: 10px;"> <div>P-A = $0,34 / 0,89 = 0,382$</div> <div>P-B = $0,27 / 0,89 = 0,303$</div> <div>P-C = $0,28 / 0,89 = 0,315$</div> </div>		
14		Con esto ya tenemos cubiertas las tres rutas posibles al destino. La siguiente hormiga al llegar en su exploración al punto P, elegirá su ruta en función a los nuevos valores de probabilidad establecidos por la cantidad de feromona depositada en los arcos correspondientes.

Tabla 27 – Evolución del valor de feromona.

• Comportamiento de la hormiga fiel:

El comportamiento de la hormiga fiel es muy simple, ya que partiendo del origen seguirá los arcos que se dirigen al destino por los puntos que marcan la menor distancia. Con esto conseguirá siempre iguales o mejores rutas que las exploradas por las hormigas de la colonia.

La hormiga fiel, partiendo de un punto determinado, busca entre todos los destinos posibles aquel que la suma de la distancia de su arco más la distancia al destino del punto al que se dirige el arco, es de menor valor. El siguiente ejemplo ayuda a la comprensión de esta toma de decisión.

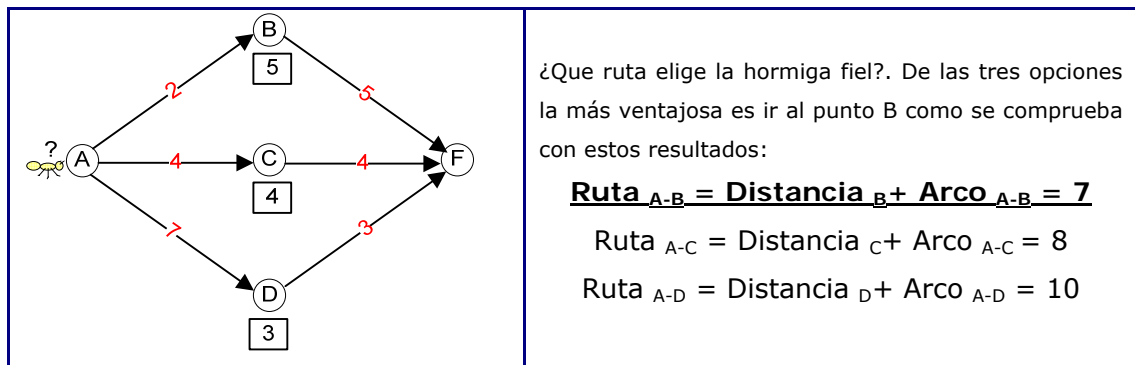


Tabla 28 – Comportamiento de la hormiga fiel.

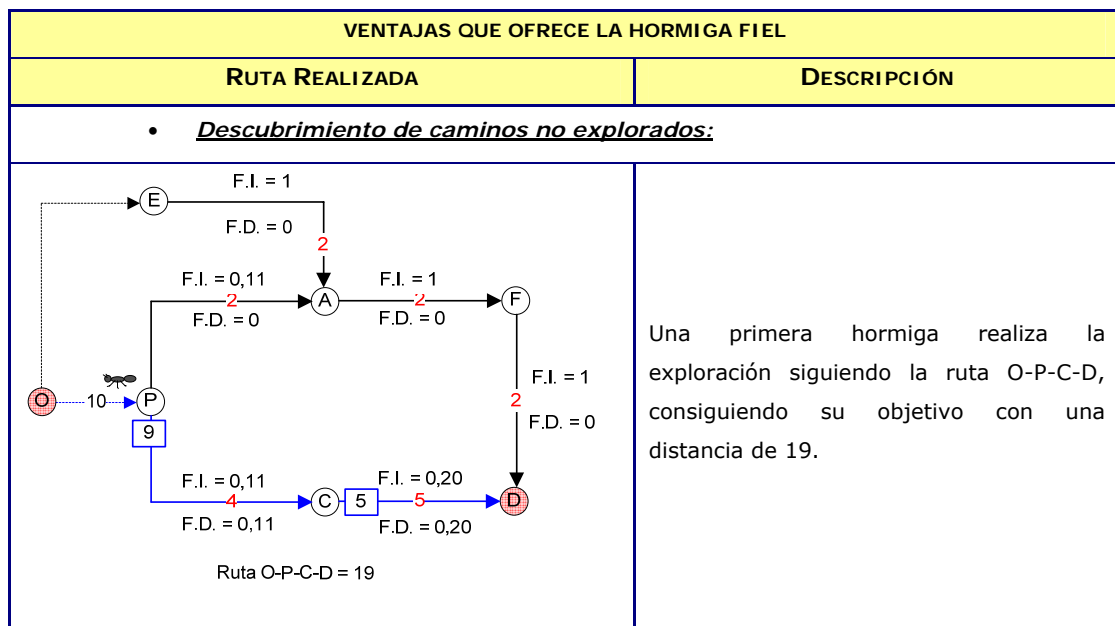
Una posible representación más formal sería la siguiente:

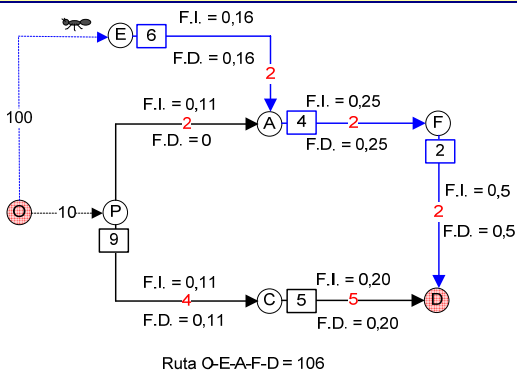
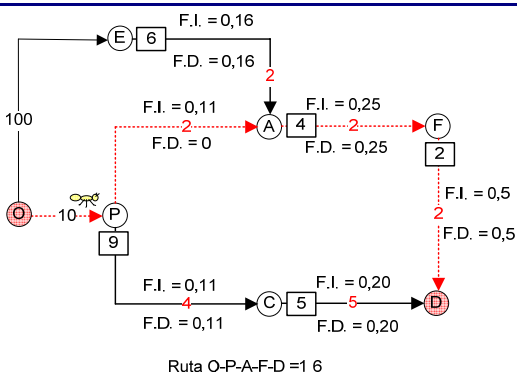
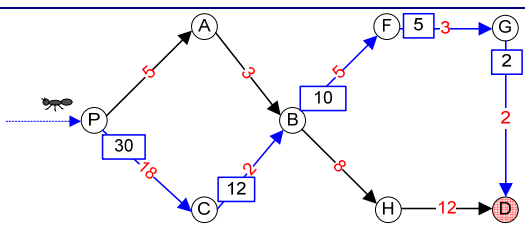
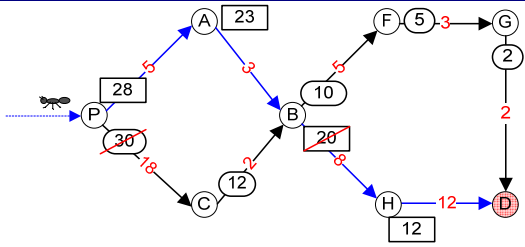
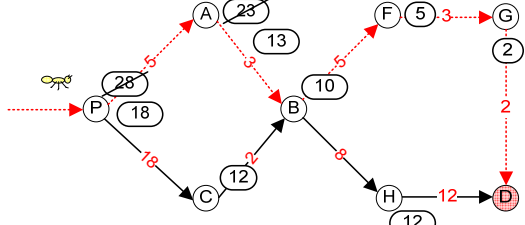
$$Punto_{Destino} \Rightarrow \text{Min}(Distancia_{i \rightarrow j} + Distancia_{j \rightarrow Destino})$$

El punto destino, elegido por la hormiga fiel, es aquel en que la suma de la distancia del arco más la distancia desde el punto fin del arco al destino final es mínima.

El comportamiento de la hormiga fiel, siguiendo este modelo, aporta una información muy importante al resultado global. Es capaz de mejorar los resultados obtenidos por la colonia de hormigas y nunca ofrecer un resultado peor que el mejor descubierto, como ocurría en modelos anteriores.

Un ejemplo completo, tanto del comportamiento de hormigas de la colonia, como de la hormiga fiel, aclara la ventaja de la utilización de esta última, por el aporte de información que realiza.



VENTAJAS QUE OFRECE LA HORMIGA FIEL	
ruta realizada	DESCRIPCIÓN
 <p>Ruta O-E-A-F-D = 106</p>	<p>Otra hormiga de la colonia realiza otra exploración siguiendo la ruta O-E-A-F-D, con una distancia de 106.</p>
 <p>Ruta O-P-A-F-D = 16</p>	<p>La hormiga fiel es capaz de encontrar una ruta con un valor menor. Para ello sigue la ruta O-P-A-F-D, con un valor de 16. Esto lo consigue sin que ningún miembro de la colonia haya realizado la exploración del arco P-A, ya que al llegar al punto P es capaz de identificar al punto A como más cercano al destino y por lo tanto que la ruta explorada desde ese punto (P-C-D).</p>
<p>• <u>Eliminación de cruces de ciclos:</u></p>	
	<p>Una hormiga recorre la ruta P-C-B-F-G-D, con una distancia de 30.</p>
	<p>Otra hormiga de la colonia realiza un recorrido diferente, pero con puntos en común, en este caso P-A-B-H-D, con una distancia de 28, mejorando la realizada por la hormiga anterior.</p>
	<p>El problema es que ninguno de los dos recorridos ofrece el mejor resultado. La hormiga fiel es capaz de identificar este cruce y encontrar la distancia mínima que ofrecen las dos rutas. En este caso P-A-B-F-G-D con una</p>



VENTAJAS QUE OFRECE LA HORMIGA FIEL	
RUTA REALIZADA	DESCRIPCIÓN
	distancia de 18.

Tabla 29 – Ventajas de la hormiga fiel.

Como se puede comprobar se ha mejorado de una manera ostensible, tanto el comportamiento de los elementos de la colonia, como la utilidad de la hormiga fiel. Este modelo, teóricamente gestiona la información de una manera más eficaz y su complejidad no ha sufrido prácticamente alteración.

3.5.3 REFINAMIENTO DE LA SOLUCIÓN ADOPTADA.

Durante el periodo de implementación se fueron realizando pruebas de comprobación de funcionamiento, rendimiento, resultados, observando un conjunto de comportamientos susceptibles de mejora. Eran pequeñas modificaciones que refinaban el comportamiento global del software desarrollado, por lo que se incluye en esta misma fase, ofreciendo los resultados con estas modificaciones ya realizadas.

Las precisiones realizadas son las que se describen a continuación en los siguientes puntos:

3.5.3.1 CENTRAR A LA COLONIA DE HORMIGAS EN EL PROBLEMA.

Utilizamos la disposición aleatoria de las hormigas en la red como optimización en la búsqueda del punto destino. Una vez marcado el objetivo y teniendo una ruta que une el origen y el destino, el disponer las hormigas de la colonia de forma aleatoria puede resultar de poca utilidad, debido a su disposición aleatoria muchas se sitúan en zonas poco prometedoras en cuanto aporte de conocimiento.

Una vez que una hormiga de la colonia consigue trazar la ruta completa entre el origen y el destino, es cuando la hormiga fiel comienza a realizar la búsqueda de la ruta óptima hasta el momento explorada. Basándose en el recorrido actual de la hormiga fiel, es decir, el mejor recorrido encontrado hasta el momento, se va a intentar centrar a la colonia en la búsqueda sobre este camino.

Como se ha indicado existen dos tipos de hormigas. El conjunto de hormigas que parten del origen no sufrirá modificación y continuará partiendo del origen. En cambio, el otro conjunto que se distribuye aleatoriamente, en su nacimiento se situarán de manera aleatoria en alguna posición correspondiente a la ruta seguida por la hormiga fiel.

Esto presenta una ventaja en redes grandes, teóricamente se centra la búsqueda como si fuera una búsqueda local y se dejan con menos probabilidades de visitar puntos alejados de la solución.

En la siguiente ilustración se intenta reflejar como al situar el nacimiento de las hormigas en la mejor ruta hasta el momento encontrada, las probabilidades de exploración sobre zonas alejadas de la solución disminuyen.

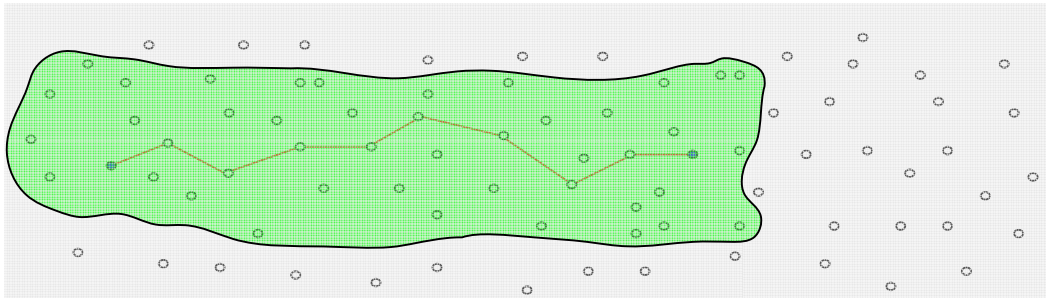


Ilustración 43 – Zona de mayor probabilidad de visita.

La probabilidad de que la población de hormigas se centre en el recorrido ya descubierto y por lo tanto que este mejore aumenta de una manera considerable con esta técnica, evitando en parte la pérdida de las hormigas por la red de puntos que forman el problema.

3.5.3.2 VALORACIÓN DE FEROMONA PARA LA TOMA DE DECISIÓN

El valor de la feromona inicial en todos los arcos es el mismo. Sirve para que todo arco tenga un valor mayor de cero, de tal manera que sea posible su exploración.

El valor de la feromona de un arco se considera la suma de la feromona inicial más la feromona depositada. Si se introduce una constante, de tal manera que influya en la valoración de una de ellas, forzará a la colonia de hormigas a seguir la ruta de mayor feromona o a la realización de una mayor exploración. Esto es lo que se ha tenido en cuenta al introducir el "*factor de feromona inicial*".

La valoración de las posibles rutas seguirá la siguiente formula:

$$F_{Arco} = F_{Inicial} \times K + F_{Depositada}$$

Siendo K un número positivo.

Si los valores de K aumentan, se da más importancia a la exploración, los valores de probabilidad para elegir un arco tienden a igualarse, al ser la feromona inicial la misma para todos los arcos. Si por el contrario, disminuimos K hacia un valor próximo a 0, la hormiga tiende a seguir aquellos caminos explorados donde la feromona depositada es mayor.

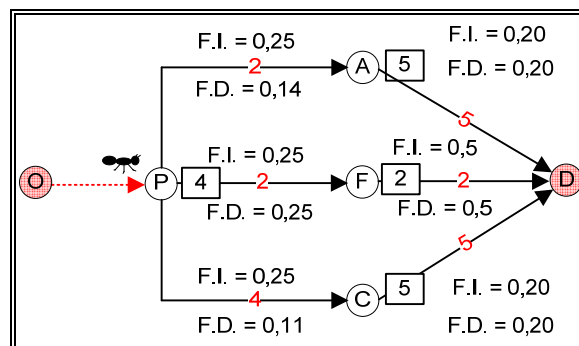


Ilustración 44 – Factor de valoración de la feromona inicial.

En las siguientes tablas se puede comprobar lo anteriormente expuesto y observar que para valores superiores a uno, las probabilidades tienden a igualarse para las tres rutas posibles de la hormiga. Si por el contrario, el factor de feromona es menor de uno, los valores de las probabilidades tienden a distanciarse.

○ **Valor Neutro: $K = 1$.**

Con este valor se deja que tenga el mismo peso tanto la feromona inicial como la depositada para la decisión del camino a elegir.

			1	
			Valor Arco	%
A	Feromona Inicial	0,25	0,390	31,200
	Feromona Depositada	0,14		
F	Feromona Inicial	0,25	0,500	40,000
	Feromona Depositada	0,25		
C	Feromona Inicial	0,25	0,360	28,800
	Feromona Depositada	0,11		

Ilustración 45 – Factor de feromona inicial. Valor de K neutro.

○ **Predomina la explotación: $K < 1$.**

Se muestra la variación de probabilidad de elección de recorrido para valores de K de 0,1 y 0,5. Con estos valores se comprueba como los valores de probabilidades de elección de los posibles arcos cambian, aumentando en el de mayor cantidad de feromona como es en el caso del punto F . Según disminuye el valor de K , las probabilidades aumentan en los arcos de mayor feromona.

			0,1		0,5	
			Valor Arco	%	Valor Arco	%
A	Feromona Inicial	0,25	0,165	28,696	0,265	30,286
	Feromona Depositada	0,14				
F	Feromona Inicial	0,25	0,275	47,826	0,375	42,857
	Feromona Depositada	0,25				
C	Feromona Inicial	0,25	0,135	23,478	0,235	26,857
	Feromona Depositada	0,11				

Ilustración 46 – Factor de feromona inicial. Valores de K menores a uno.

○ **Predomina la exploración: $K > 1$.**

Al contrario del caso anterior cuando aumentamos el valor de K favorecemos la exploración de la red. Este valor origina un equilibrio mayor en las diferentes probabilidades que se originan en los arcos. Cuanto mayor es el valor de K , los valores de probabilidades tienden a igualarse.



			1,5		2	
			Valor Arco	%	Valor Arco	%
A	Feromona Inicial	0,25	0,515	31,692	0,640	32,000
	Feromona Depositada	0,14				
F	Feromona Inicial	0,25	0,625	38,462	0,750	37,500
	Feromona Depositada	0,25				
C	Feromona Inicial	0,25	0,485	29,846	0,610	30,500
	Feromona Depositada	0,11				

Ilustración 47 – Factor de feromona inicial. Valores de K superiores a uno.

Esta variable permite jugar con la exploración y la explotación del conocimiento.

3.5.3.3 APORTE DE FEROMONA DE LA HORMIGA FIEL

Como se ha expuesto, la hormiga fiel puede descubrir rutas mejores que las elaboradas por las hormigas de la colonia.

La hormiga fiel no solamente recorrerá su camino, sino que además efectuará, en su vuelta, un aporte de feromona a la red al igual que el resto de las hormigas de la colonia. Con esto, lo que se logra es marcar el mejor recorrido con feromona, es por lo que se puede considerar una marca elitista para su toma en consideración por el conjunto de las hormigas de la colonia.

3.5.3.4 ELIMINACIÓN DE LAS HORMIGAS CONSIDERADAS PERDIDAS.

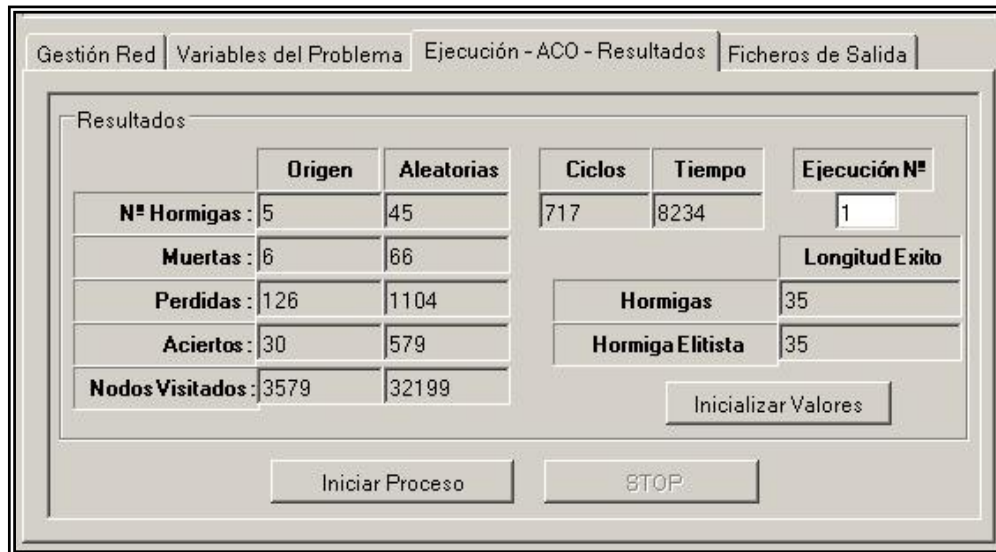
A pesar de centrar el nacimiento de las hormigas sobre la ruta, se ha probado a introducir un factor límite de supervivencia de las hormigas. En este punto se toma como inspiración otro comportamiento natural de la hormiga *Cataglyphis fortis*, por el cual debido a las altas temperaturas existentes en el medio, tiene un cierto tiempo de estancia fuera del hormiguero, falleciendo si este se prolonga. Se ha introducido una variable ("*Factor de Hormiga Perdida*", *FHP*), que constituye el valor que origina la declaración de una hormiga como perdida y por lo tanto como muerta.

Basándose en la mejor distancia hasta el momento encontrada, distancia de la hormiga fiel, se ha decidido eliminar aquellas hormigas que superan esta cantidad de una manera relevante, ya que se consideran perdidas en la red.

Una vez que la hormiga realiza su movimiento, si la distancia recorrida por la hormiga supera la mejor longitud hasta el momento encontrada tantas veces como el valor de la variable (*FHP*), se considera a la hormiga *perdida* y se decide eliminarla. Los resultados confirman que la cantidad de hormigas que se pierden es mucha, por lo que se sigue realizando exploración de la red.

$$Hormiga_Perdida \Rightarrow hormiga_{Distancia} > hormiga_fiel_{Distancia} * FHP$$

Para una red de 100 puntos, dos arcos por punto, 50 hormigas, un factor de hormiga perdida de 3 y un factor de feromona inicial de 1, como se observa en la siguiente ilustración, el número de hormigas consideradas perdidas es elevado.



The screenshot shows a software window titled 'Ejecución - ACO - Resultados'. It contains several tabs: 'Gestión Red', 'Variables del Problema', 'Ejecución - ACO - Resultados' (selected), and 'Ficheros de Salida'. The main area displays results for 'Resultados' with the following data:

	Origen	Aleatorias	Ciclos	Tiempo	Ejecución Nº
Nº Hormigas :	5	45	717	8234	1
Muertas :	6	66			
Perdidas :	126	1104			
Aciertos :	30	579			
Nodos Visitados :	3579	32199			

Additional metrics and controls:

- Longitud Exitos:** 35
- Hormigas:** 35
- Hormiga Elitista:** 35
- Inicializar Valores:** Button
- Iniciar Proceso:** Button
- STOP:** Button

Ilustración 48 – Valoración de Hormigas Perdidas.

Esto viene a indicar que aunque se centre mucho la exploración a zonas prometedoras, las hormigas siguen teniendo cierta libertad a la hora de elegir rutas distintas a las marcadas por la mayor cantidad de feromona.

3.5.4 RESULTADOS.

El objetivo principal del proyecto es comprobar el comportamiento del algoritmo para redes de tamaño grande. Existe un conjunto de variables que influyen en los resultados proporcionados por el programa, por este motivo se ha intentado determinar en un primer momento, que valores, de las diferentes variables, pueden presentar mejores resultados. Una vez finalizado este paso, comprobar hasta que tamaño los resultados ofrecidos son aceptables es un paso trivial.

Hay que mencionar que lo que se busca con este tipo de algoritmos es una respuesta ágil y aceptable, no se busca el óptimo, pero si un valor cercano en un tiempo aceptable.

Antes de la realización de estas comprobaciones sobre las variables, se ha realizado una estimación del tamaño del problema capaz de resolver el algoritmo y sobre este tamaño se han efectuado estas comprobaciones.

Para ver el comportamiento general se han realizado un conjunto de pruebas sobre las variables que intervienen en su resolución. Los experimentos que a continuación se exponen de una manera más detallada son los siguientes:

- [Análisis del número de hormigas de la colonia.](#) Determinación del número de hormigas que forman la colonia que mejor resultado aporta a los experimentos.
- [Análisis porcentaje en origen de las hormigas.](#) Determinación del porcentaje de hormigas de cada grupo, aquellas que se posicionan de

manera aleatoria por la red y las que parten del punto origen, que mejor resultado ofrecen.

- Análisis factor de feromona inicial. Determinación del valor de K que interviene como factor de la feromona inicial que mejor resultado aporta en la ejecución del algoritmo.
- Análisis de dimensiones de la red. Una vez determinadas las variables anteriormente indicadas, se intenta realizar el estudio que marca el objetivo prioritario del proyecto, la determinación de las dimensiones de la red en las que el algoritmo presenta unos resultados aceptables.

3.5.4.1 ANÁLISIS NÚMERO DE HORMIGAS DE LA COLONIA

Se ha intentado determinar el número de hormigas integrantes de la colonia que mejores resultados aporta al algoritmo desarrollado. Para ello se han realizado un conjunto de pruebas que faciliten información y permitan determinar el valor más fiable que origine los mejores resultados en los cálculos de dimensionamiento de la red.

Se han realizado las pruebas para varios tamaños de redes, con una complejidad similar al crearse de manera aleatoria todos los conjuntos. Las colonias de hormigas varían su número desde 10 hasta 200, obteniendo los siguientes resultados:

- Conjunto de resultados para 500 puntos IV R500-NH:

ENTORNO EXPERIMENTAL	
<ul style="list-style-type: none"> Se genera una red de 500 puntos. 	
<ul style="list-style-type: none"> Cada punto tiene como origen 2 arcos, y como destino de cada arco un punto aleatorio de la red. 	
<ul style="list-style-type: none"> Se muestran los resultados para pruebas de 10, 50, 100 y 200 hormigas en la colonia. 	
<ul style="list-style-type: none"> El valor de feromona inicial es de 1. 	
<ul style="list-style-type: none"> Porcentaje de hormigas en origen 50%. 	
<ul style="list-style-type: none"> Factor de feromona inicial $K=1$. 	

Tabla 30 – Entorno experimental. Cálculo número hormigas. IV_R500-NH.

500															
10				50				100				200			
Colonia		Fiel		Colonia		Fiel		Colonia		Fiel		Colonia		Fiel	
Distancia	Tiempo	Distancia	Tiempo	Distancia	Tiempo	Distancia	Tiempo	Distancia	Tiempo	Distancia	Tiempo	Distancia	Tiempo	Distancia	Tiempo
366	1468	366	1484	145	1500	145	1531	189	750	189	766	185	687	93	718
352	2250	101	2250	45	3421	143	1640			140	1047	45	1890	69	812
108	11375	39	3468	39	4640	95	1921			39	1110	39	4672	61	1031
90	14531					80	3000							39	1906
45	29078					39	3421								
39	38578														

Ilustración 49 – Resultados. Cálculo número hormigas. IV_R500-NH.

Como se observa en la tabla se muestran los datos para 500 puntos, con un número de hormigas de 10, 50, 100 y 200, mostrando los resultados de la distancia y el tiempo transcurrido en milisegundos, tanto de la mejor hormiga de la colonia como de la hormiga fiel.

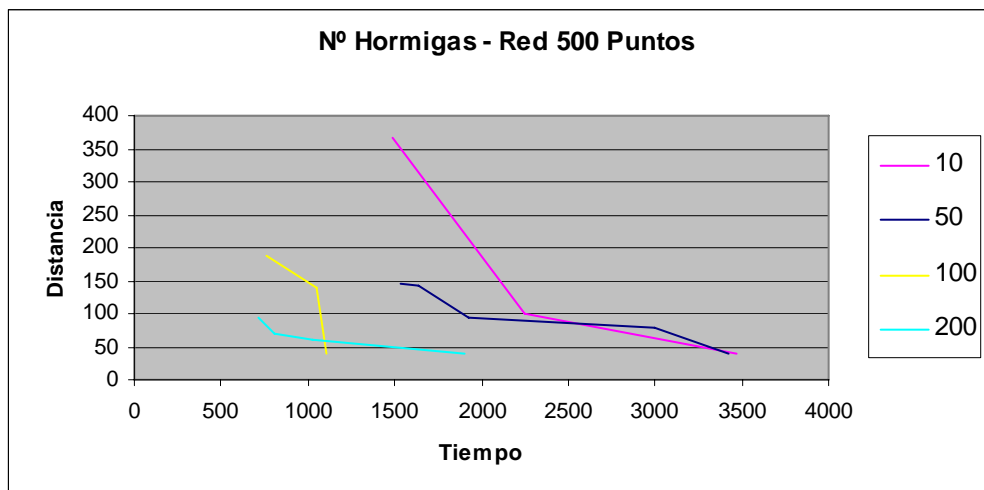


Ilustración 50 – Gráfico. Cálculo número hormigas. IV_R500-NH.

o Conclusión:

Como se puede comprobar, los mejores resultados marcan a dos conjuntos, los formados con 100 y 200 hormigas. Hay que indicar que si el tamaño de la red aumenta, teóricamente para mantener la misma exploración, también habría que incrementar el número de hormigas. Este aumento también puede ser contraproducente, el aumentar el número de hormigas obliga a la penalización de tiempo en cada ciclo, ya que el movimiento se realiza sobre un conjunto más grande de población.

Se puede observar como el conjunto de 200 hormigas encuentra una mejor solución en un primer momento pero el de cien es capaz de mejorarla en menos de medio segundo, tardando el resto de conjuntos casi tres segundos más para conseguirlo.

- Conjunto de resultados para 1000 puntos. IV_R1000-NH.

ENTORNO EXPERIMENTAL	
• Se genera una red de 1000 puntos.	
• Cada punto tiene como origen 2 arcos y como destino de cada arco un punto aleatorio de la red.	
• Se muestran los resultados para pruebas de 10, 50, 100 y 200 hormigas en la colonia.	
• El valor de feromona inicial es de 1.	
• Porcentaje de hormigas en origen 50%.	
• Factor de feromona inicial K=1.	

Tabla 31 – Entorno experimental. Cálculo número hormigas. IV_R1000-NH.

1000															
10				50				100				200			
Colonia	Fiel	Colonia	Fiel	Colonia	Fiel	Colonia	Fiel	Colonia	Fiel	Colonia	Fiel	Colonia	Fiel	Colonia	Fiel
Distancia	Tiempo	Distancia	Tiempo	Distancia	Tiempo	Distancia	Tiempo	Distancia	Tiempo	Distancia	Tiempo	Distancia	Tiempo	Distancia	Tiempo
186	6765	186	6812	337	687	227	719	326	953	125	969	236	844	231	891
125	25594	115	7844	218	2859	116	1422	110	1562	105	1531	114	3297	116	969
		101	8687	186	3844	114	2828			76	1765	112	3391	109	2859
		91	21125	130	8109	98	3062					103	3844	76	3375
				124	21734	96	5094								
				99	26859	85	7375								
				92	27422	81	17234								
						73	26422								

Ilustración 51 – Resultados. Cálculo número hormigas. IV_R1000-NH.

En la ilustración anterior se muestran los datos para 1000 puntos, con un número de hormigas de 10, 50, 100 y 200, mostrando los resultados de la distancia y el tiempo transcurrido en milisegundos, tanto de la mejor hormiga de la colonia como de la hormiga fiel.

Los resultados representados de manera gráfica son los siguientes:

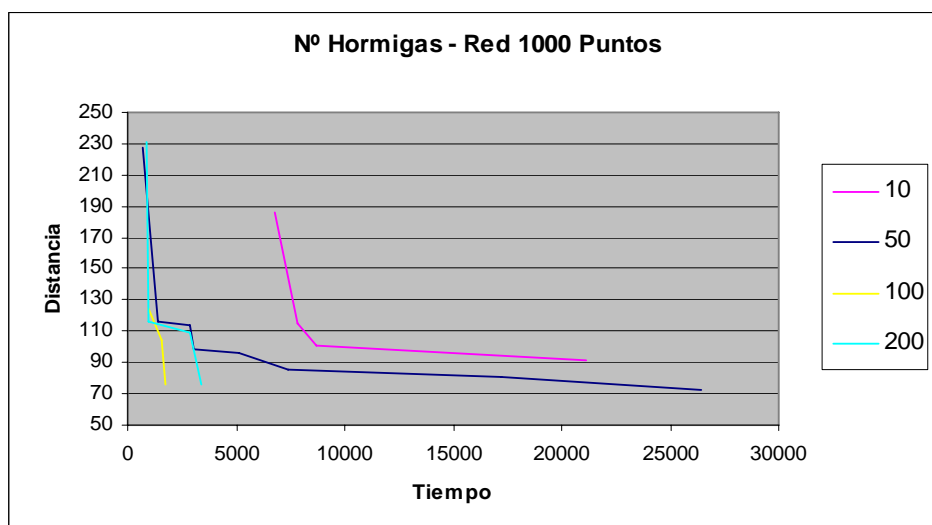


Ilustración 52 – Gráfico. Cálculo número hormigas. IV_R1000-NH.



○ Conclusión:

Como en el caso anterior, el conjunto que presenta mejores resultados es el de 100 y 200 hormigas, aunque en este caso la colonia de 50 también obtiene buenos resultados pero con un coste de tiempo mayor.

Hay que indicar que la colonia de 10 hormigas queda completamente desfasada del resultado del resto de colonias.

• Conjunto de resultados para 2000 puntos. IV_R2000-NH

ENTORNO EXPERIMENTAL	
• Se genera una red de 2000 puntos.	
• Cada punto tiene como origen 2 arcos y como destino de cada arco un punto aleatorio de la red.	
• Se muestran los resultados para pruebas de 10, 50, 100 y 200 hormigas en la colonia.	
• El valor de feromona inicial es de 1.	
• Porcentaje de hormigas en origen 50%.	
• Factor de feromona inicial $K=1$.	

Tabla 32 – Entorno experimental. Cálculo número hormigas. IV_R2000-NH.

2000															
10				50				100				200			
Colonia		Fiel		Colonia		Fiel		Colonia		Fiel		Colonia		Fiel	
Distancia	Tiempo	Distancia	Tiempo	Distancia	Tiempo	Distancia	Tiempo	Distancia	Tiempo	Distancia	Tiempo	Distancia	Tiempo	Distancia	Tiempo
				414	938	181	985	217	656	211	703	183	797	183	828
				202	1625	162	1981	194	2578	193	1469	181	10453	161	1750
				182	2578	161	2594	175	15500	191	1562	176	16781	143	4984
				178	19969	159	3031	161	35141	127	3844	175	20156	138	14578
				171	25110	157	3297	159	55234	125	35156	165	28172		
				169	227266	154	8406					157	22828		
				167	297688	148	60938					154	79266		
						147	83141								
						136	120281								

Ilustración 53 – Resultados. Cálculo número hormigas. IV_R2000-NH.

En la ilustración anterior se muestran los datos para 2000 puntos, con un número de hormigas de 50, 100 y 200, mostrando los resultados de la distancia y el tiempo transcurrido en milisegundos, tanto de la mejor hormiga de la colonia como de la hormiga fiel.

Los resultados de manera gráfica son los siguientes:

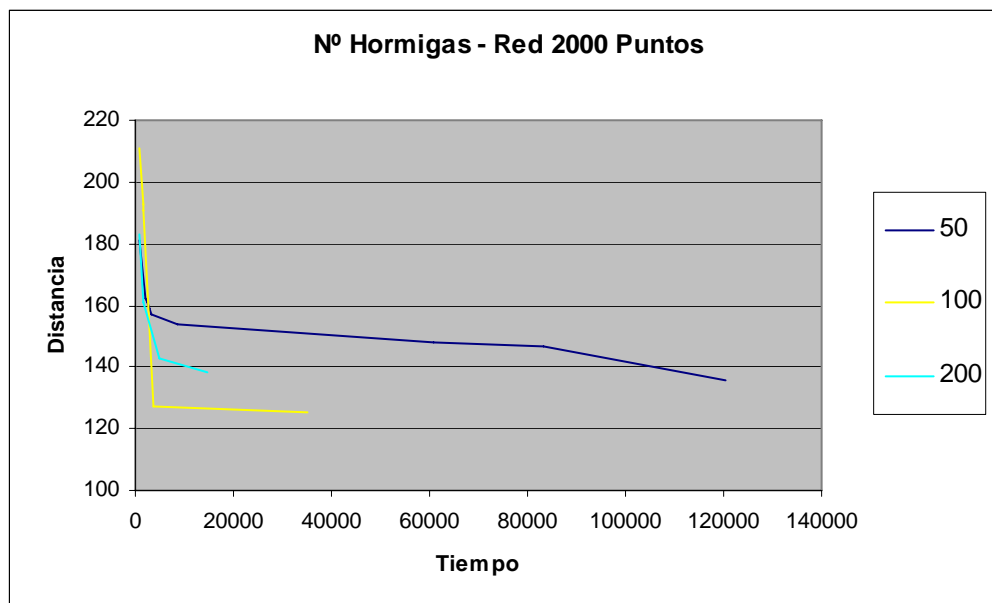


Ilustración 54 – Gráfico. Cálculo número hormigas. IV_R2000-NH.

○ Conclusión:

Con este nuevo tamaño se aumenta de una manera considerable la complejidad. Se ha decidido prescindir de los resultados del conjunto de 10 hormigas, ya en el tamaño analizado anteriormente ya presentaba unos resultados bastante malos en comparación con el resto de conjuntos.

En esta prueba se viene a confirmar lo expuesto, y ya el grupo de 50 hormigas también queda de una manera evidente la peor clasificada. En cambio el grupo de 100 hormigas sigue siendo el que mejores resultados aporta.

• Conjunto de resultados para 5000 puntos IV_R5000-NH.

ENTORNO EXPERIMENTAL	
•	Se genera una red de 5000 puntos.
•	Cada punto tiene como origen 2 arcos y como destino de cada arco un punto aleatorio de la red.
•	Se muestran los resultados para pruebas de 50, 100 y 200 hormigas en la colonia.
•	El valor de feromona inicial 1.
•	Porcentaje de hormigas en origen 50%.
•	Factor de feromona inicial $K=1$.

Tabla 33 – Entorno experimental. Cálculo número hormigas. IV_R5000-NH.



Recorridos de grafos mediante hormigas.

5000															
10				50				100				200			
Colonia		Fiel		Colonia		Fiel		Colonia		Fiel		Colonia		Fiel	
Distancia	Tiempo	Distancia	Tiempo	Distancia	Tiempo	Distancia	Tiempo	Distancia	Tiempo	Distancia	Tiempo	Distancia	Tiempo	Distancia	Tiempo
				1043	7343	439	7390	459	1344	434	1407	1202	4641	464	4688
				513	33422	437	9359			414	7844	538	7625	458	7063
				500	86172	434	16640			408	7907	532	10438	440	7641
						427	33968			399	15860	522	22422	439	9829
						423	90093			397	78094			436	12922
														435	14454
														429	14922
														415	35219

Ilustración 55 – Resultados. Cálculo número hormigas. IV_R5000-NH.

En la ilustración anterior se muestran los datos para 5000 puntos, con un número de hormigas de 50, 100 y 200, mostrando los resultados de la distancia y el tiempo transcurrido en milisegundos, tanto de la mejor hormiga de la colonia como de la hormiga fiel.

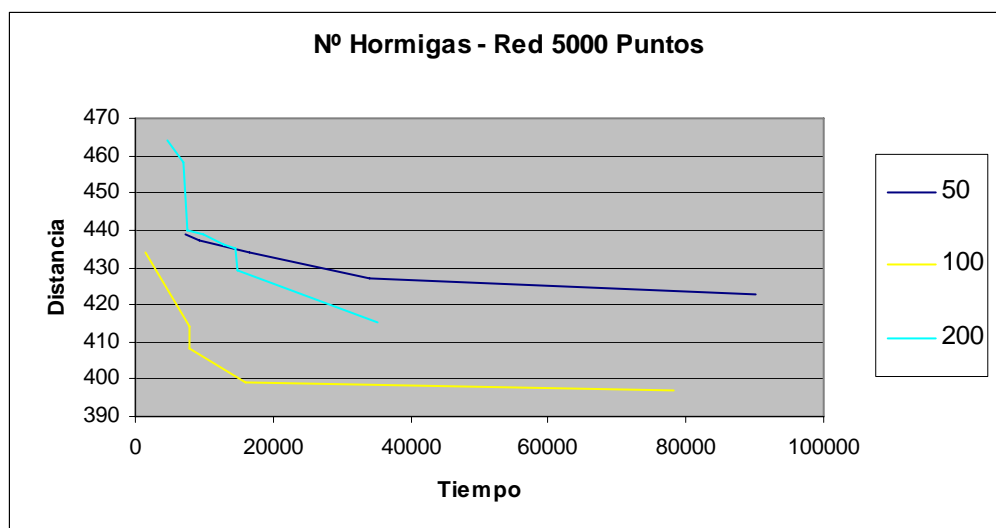


Ilustración 56 – Gráfico. Cálculo número hormigas. IV_R5000-NH.

○ Conclusión:

Como en el caso anterior, queda en evidencia que aunque se aumente el número de puntos, se sigue manteniendo en cabeza, con unos resultados aventajados, la colonia de 100 hormigas.

Con el conjunto de pruebas expuestas, se considera que para los tamaños de red que se han considerado objeto de estudio de este proyecto, el número de hormigas de la colonia que mejores resultados puede aportar es de 100.

Nº de Hormigas de la Colonia = 100



3.5.4.2 ANÁLISIS PORCENTAJE EN ORIGEN DE LAS HORMIGAS

De las variables introducidas en el algoritmo, esta hace referencia al número de las hormigas que parten del origen en relación con aquellas que se sitúan de forma aleatoria por la red. Para ver que valor presenta unos mejores resultados, se han realizado un conjunto de pruebas con diferentes valores, que varían del 10% hasta el 60% para un tamaño de red de 5000 puntos, que se considera de una extensión adecuada para el análisis a desarrollar en este proyecto.

ENTORNO EXPERIMENTAL	
•	Se genera una red de 5000 puntos.
•	Cada punto tiene como origen 2 arcos y como destino de cada arco un punto aleatorio de la red.
•	Se muestran los resultados para pruebas de 10%, 20%, 30%, 40%, 50% y 60%, de porcentaje de hormigas que parten del punto origen del trayecto.
•	El valor de feromona inicial es de 1.
•	Número de hormigas de la colonia 100.
•	Factor de feromona inicial $K=1$.

Tabla 34 – Entorno experimental. Cálculo porcentaje en origen. IV_R5000-PO.

Las tablas siguientes muestran los datos de los resultados obtenidos, informando del ciclo y tiempo en milisegundos donde ha ocurrido el evento y de la distancia menor obtenida hasta ese momento tanto de la hormiga fiel (*DHF*, hormiga que recorre los puntos siguiendo el valor máximo de feromona), como de las hormigas de la colonia (*DHC*).

5000											
10%				20%				30%			
Ciclo	Tiempo	DHF	DHC	Ciclo	Tiempo	DHF	DHC	Ciclo	Tiempo	DHF	DHC
102	1437	573	584	198	2843	573	698	97	1344	573	573
124	2140	514	584	467	11922	545	698	193	4297	523	573
151	2969	490	584	876	25656	536	698	271	6734	490	573
269	6609	478	584	911	26765	524	645	1203	29937	488	506
1922	57251	433	531	921	27140	471	645	1320	32406	483	506
2305	68423	430	531	945	27969	456	645	1588	38125	482	506
3241	95970	429	531	1419	43890	441	603	1926	45328	481	506
4405	130345	422	476	2088	65811	439	502	2458	56531	478	506
				3111	95699	427	502	2770	63140	455	506
				3121	95965	422	502	3332	75030	445	506
				5481	154727	420	454	6900	177045	441	470
				5715	160586	414	454	6968	179138	431	470
				14144	373917	412	454	9321	242263	428	470

Ilustración 57 – Resultados. Cálculo porcentaje en origen. IV_R5000-PO-I.

5000											
40%				50%				60%			
Ciclo	Tiempo	DHF	DHC	Ciclo	Tiempo	DHF	DHC	Ciclo	Tiempo	DHF	DHC
89	1234	528	535	99	1391	549	549	100	1375	523	523
218	5094	522	535	103	1516	546	549	177	3782	496	523
686	18922	504	535	197	4485	531	531	500	13438	487	523
727	20156	499	535	210	4891	529	531	726	20407	484	523
911	25797	497	535	378	10125	491	515	859	24501	477	523
1730	50797	478	530	574	16235	490	515	1056	30111	466	523
2687	80470	462	530	962	28813	436	476	1927	52909	453	500
2763	82767	440	530	2603	78189	411	473	2094	57128	428	500
3137	94032	431	502	3992	120220	404	473	2733	73082	427	491
3302	98970	430	502	4910	147752	397	473	4209	110865	426	489
3491	104626	428	502	5349	160955	396	457	5807	152117	418	489
4692	140314	423	502	8988	270004	395	457	6781	176883	406	489
5198	155533	422	502	9339	280598	393	457	17489	459578	399	471
6183	185096	406	502	11866	349552	388	457				
6965	208628	399	485	12824	372958	383	457				
				25632	687759	381	457				

Ilustración 58 – Resultados. Cálculo porcentaje en origen. IV_R5000-PO-II.

Los resultados los aporta el fichero creado por la aplicación denominado "NOTAS_Hormiga_Resultados.txt". Un análisis de los resultados se realiza mejor mediante gráficos, ya que la observación visual presenta una comparación de los resultados de una manera global más comprensible.

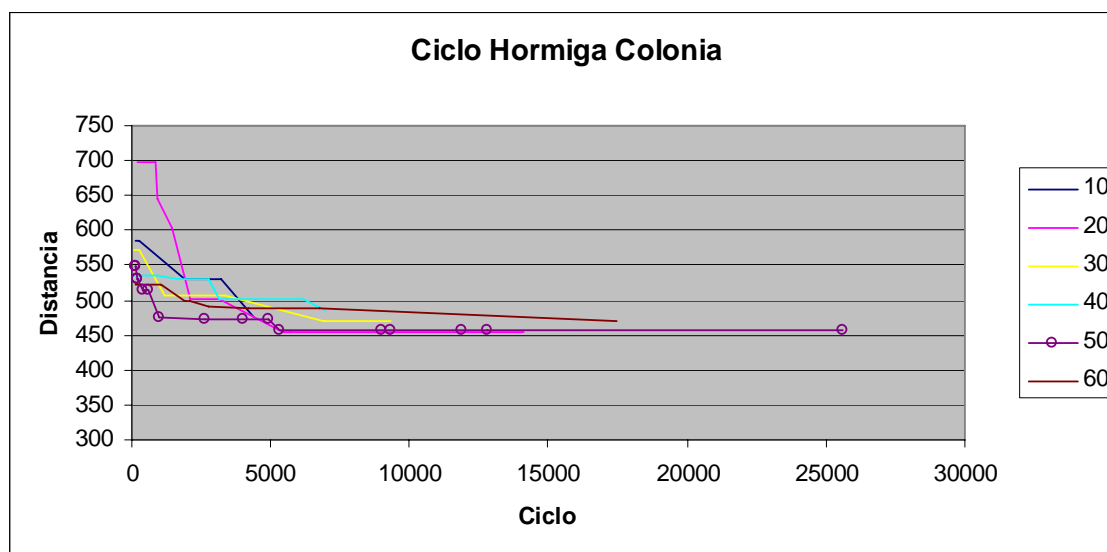


Ilustración 59 – Gráfico. Cálculo porcentaje en origen. IV_R5000-PO-HC.

Se puede observar como los diferentes conjuntos encuentran la primera solución en ciclos muy próximos y como progresan encontrando mejores soluciones según avanzan en número de ciclos. Hay que destacar que el mejor resultado lo aporta el conjunto del 50% (una distribución equilibrada entre los dos tipos de hormigas de la colonia), que parece destacar del resto de los conjuntos de prueba. Es el que antes ofrece una solución cercana a la mejor conseguida.

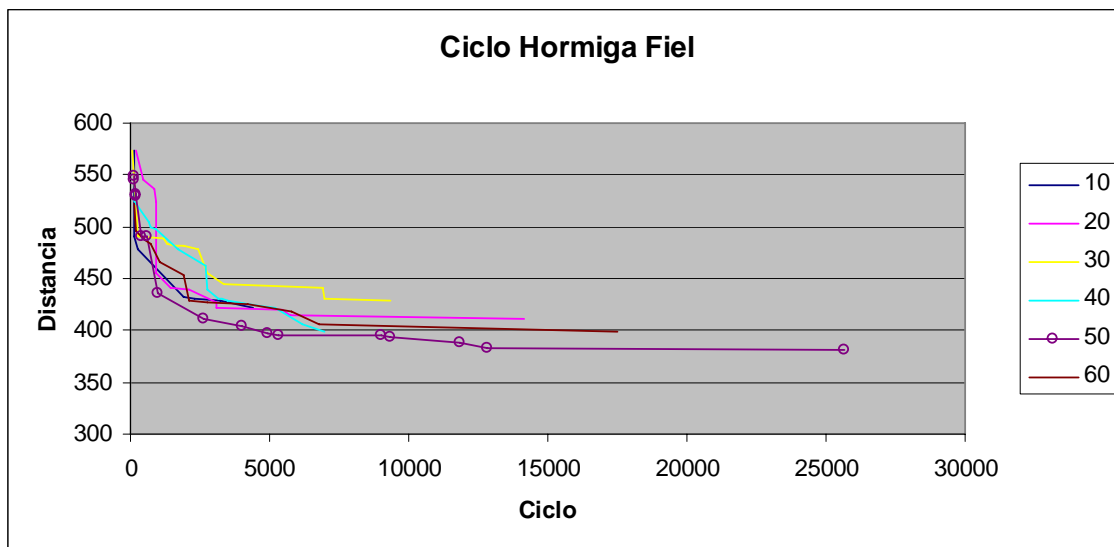


Ilustración 60 – Gráfico. Cálculo porcentaje en origen. IV_R5000-PO-HF.

En el caso de la hormiga fiel sucede un comportamiento similar al de la colonia, pero observando quizás un avance más rápido en la obtención de mejores resultados, guiando por tanto a las hormigas de la colonia hacia mejores rutas.

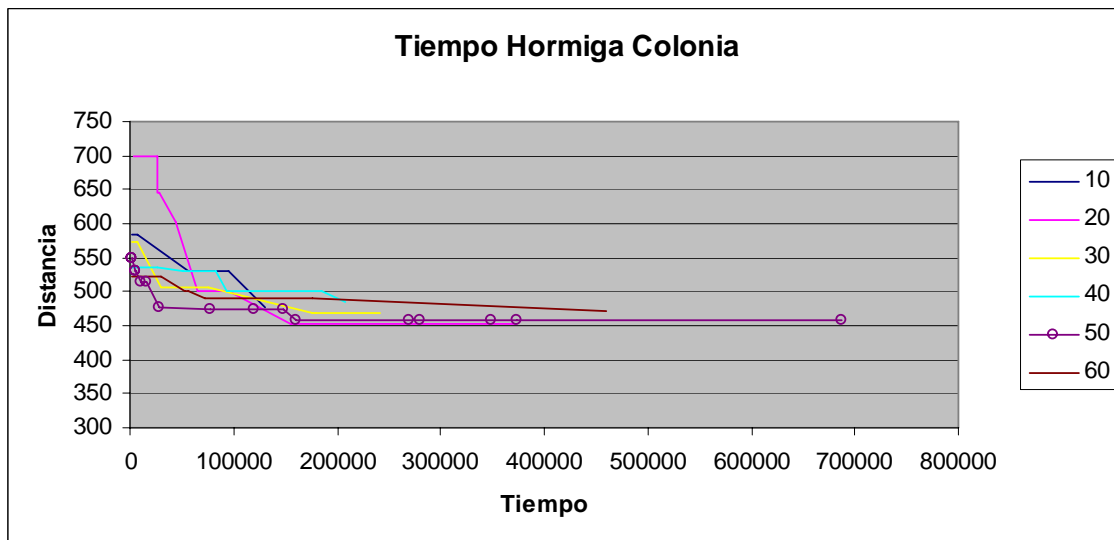


Ilustración 61 – Gráfico. Cálculo porcentaje en origen. IV_R5000-PO-HCT.

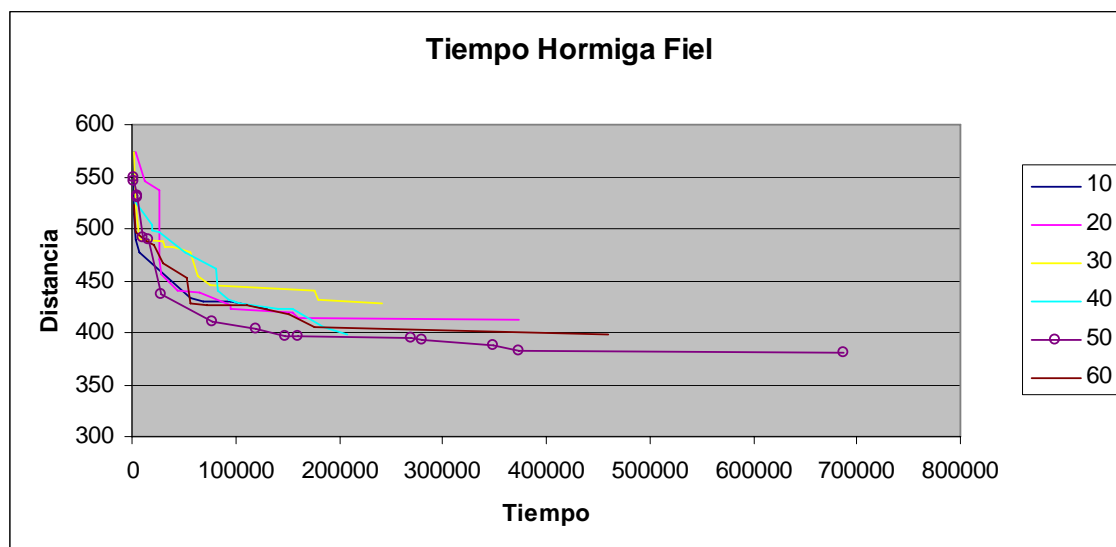


Ilustración 62 – Gráfico. Cálculo porcentaje en origen. IV_R5000-PO-HFT.

En cuanto a los resultados en tiempo son semejantes al de número de ciclos, ya que son directamente proporcionales.

○ Conclusión:

Con lo anteriormente expuesto, se considera que la colonia de hormigas puede repartirse equitativamente entre los dos conjuntos de hormigas que se crean en el algoritmo. Así un 50% puede partir del origen y el resto distribuirse aleatoriamente, sirviendo de guía al primer conjunto en la obtención del objetivo en un primer momento, y posteriormente centrarse en la mejora de la ruta elegida.

Porcentaje de Hormigas en Origen = 50%

3.5.4.3 ANÁLISIS FACTOR DE FEROMONA INICIAL.

Como en los dos puntos anteriores, otra variable que interviene en la ejecución del algoritmo es el factor de feromona inicial. Este marca la influencia de la cantidad de feromona inicial en el cálculo de probabilidades para el desplazamiento de una hormiga. El algoritmo realiza una suma de las dos cantidades de feromona que presenta un posible arco de desplazamiento. La feromona inicial presenta para todos los posibles arcos un mismo valor, es decir si doy importancia a este valor realizo una mayor exploración, ya que las probabilidades de los arcos tienden a igualarse. Si por el contrario, realizo la función contraria, doy más probabilidad al mejor recorrido encontrado desde ese punto.

Lo que se trata de averiguar es que valor puede dar un mejor resultado para la resolución de nuestro problema, buscando un equilibrio entre la exploración y la explotación del conocimiento, tomando como base la complejidad, en cuanto a dimensiones de red, estimada de estudio. Si la complejidad de la red cambia de manera ostensible, este valor se tendría que recalcular para proporcionar unos mejores resultados.

La formula utilizada para el cálculo de la feromona de un arco es la siguiente

$$F_{Arco} = F_{Inicial} * K + F_{Depositada}$$

Se realiza un estudio de diversos valores de K, que varían en el intervalo de 0,6 a 1,4, para comprobar cual de ellos obtiene unos mejores resultados. Se toma como complejidad para la realización de la pruebas una red de 1000 puntos, con 2 arcos por punto.

ENTORNO EXPERIMENTAL	
•	Se genera una red de 1000 puntos.
•	Cada punto tiene como origen 2 arcos y como destino de cada arco un punto aleatorio de la red.
•	Se muestran los resultados para pruebas con valores de 0,6, 0,8, 1,0, 1,2 y 1,4 de valores de K, factor de feromona inicial.
•	El valor de feromona inicial es de 1.
•	Número de hormigas de la colonia 100.
•	Porcentaje de hormigas en el origen del 50%.

Tabla 35 – Entorno experimental. Cálculo del factor de feromona inicial. IV_R1000-FI.

Las tablas siguientes muestran los datos de los resultados obtenidos, informando del ciclo y tiempo en milisegundos donde ha ocurrido el evento y de la distancia menor obtenida hasta ese momento tanto de la hormiga fiel (DHF, hormiga que recorre los puntos siguiendo el valor máximo de feromona), como de las hormigas de la colonia (DHC), en un entorno de pruebas de 1000 puntos para diferentes valores del factor de feromona inicial K, anteriormente explicado.

1000											
K = 0,6				K = 0,8				K = 1			
Ciclo	Tiempo	DHF	DHC	Ciclo	Tiempo	DHF	DHC	Ciclo	Tiempo	DHF	DHC
15	328	93	93	19	359	108	108	17	375	89	93
72	1172	92	93	36	625	81	108	47	828	87	93
89	1422	90	93	39	672	78	93	59	1000	81	93
109	1734	79	79	120	1859	66	86	65	1125	75	93
225	3422	66	79					236	3656	66	79

Ilustración 63 – Resultados. Cálculo del factor de feromona inicial. IV_R1000-FI-I.

1000							
K = 1,2				K = 1,4			
Ciclo	Tiempo	DHF	DHC	Ciclo	Tiempo	DHF	DHC
17	359	107	107	18	328	102	102
21	453	95	95	64	1078	84	102
23	484	93	95	90	1469	82	90
51	922	90	95	617	9125	69	69
71	1219	80	80	1060	15563	66	66
96	1609	79	80				
133	2141	73	75				
454	6906	70	75				
1426	21188	66	66				

Ilustración 64 – Resultados. Cálculo del factor de feromona inicial. IV_R1000-FI-II.

Una representación gráfica puede ayudar a una mejor interpretación de los datos obtenidos del conjunto de pruebas realizadas.

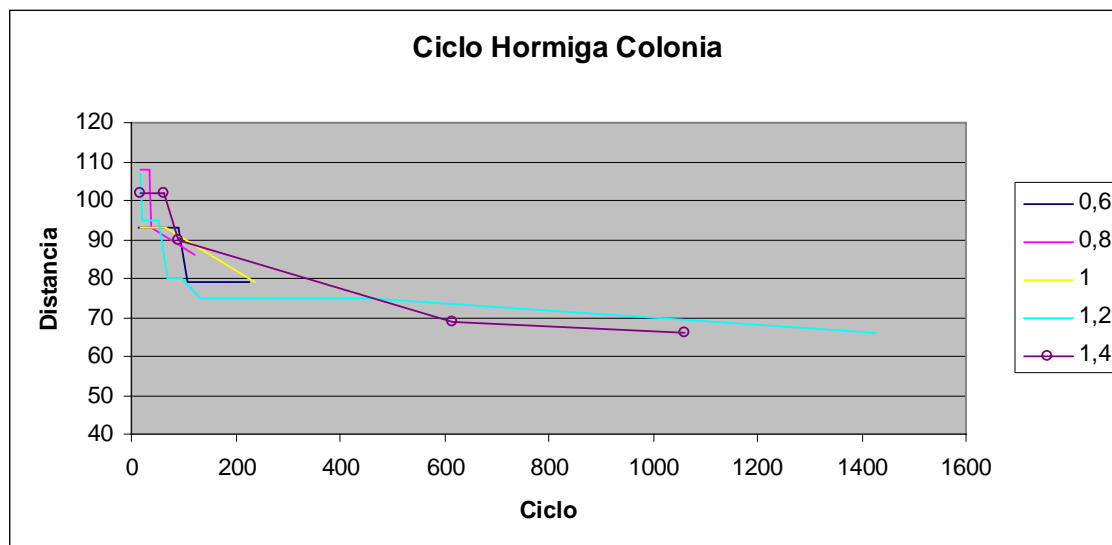


Ilustración 65 – Gráfico. Cálculo del factor de feromona inicial. IV_R1000-FI. Ciclo Hormiga Colonia.

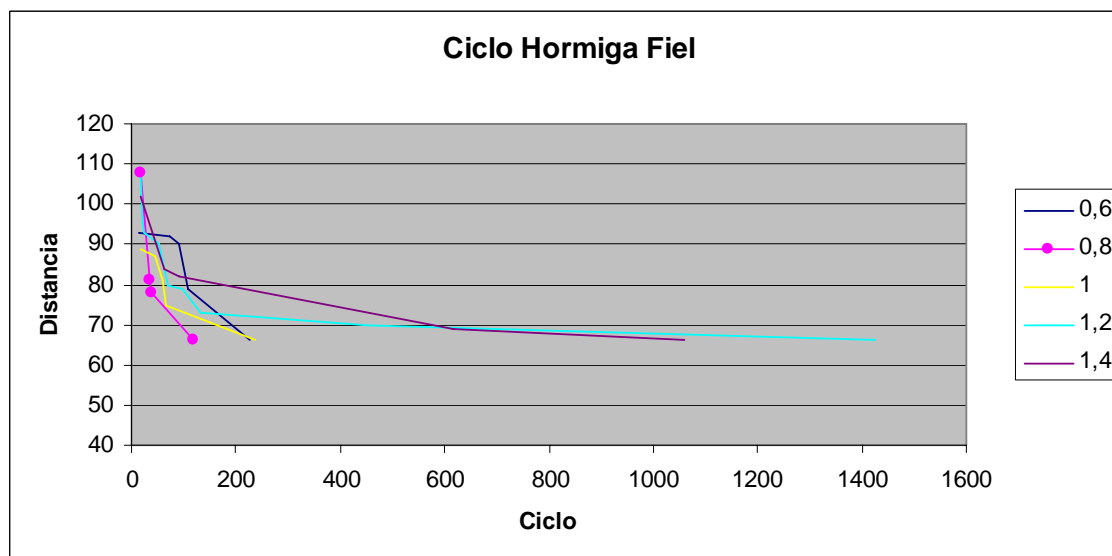


Ilustración 66 – Gráfico. Cálculo del factor de feromona inicial. IV_R1000-FI. Ciclo Hormiga Fiel.

En la observación de los dos gráficos anteriores se puede comprobar como el valor de 0,8 consigue los mejores resultados con la hormiga fiel, ya que en poco tiempo consigue el valor mínimo. En cambio para la colonia de hormigas los mejores resultados los proporciona el valor de 1,4, es decir, cuando aumento la exploración. Tarda un mayor tiempo, pero la exploración consigue en la colonia mejorar los resultados, en cambio en valores pequeños, se queda lejos de los resultados de la hormiga fiel.

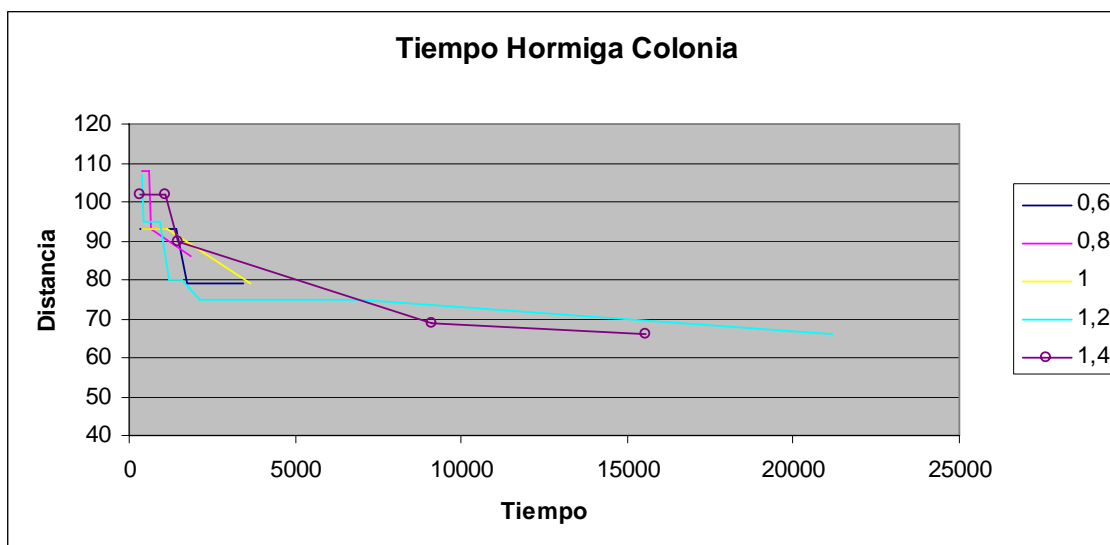


Ilustración 67 – Gráfico. Cálculo del factor de feromona inicial. IV_R1000-FI. Tiempo Hormiga Colonia.

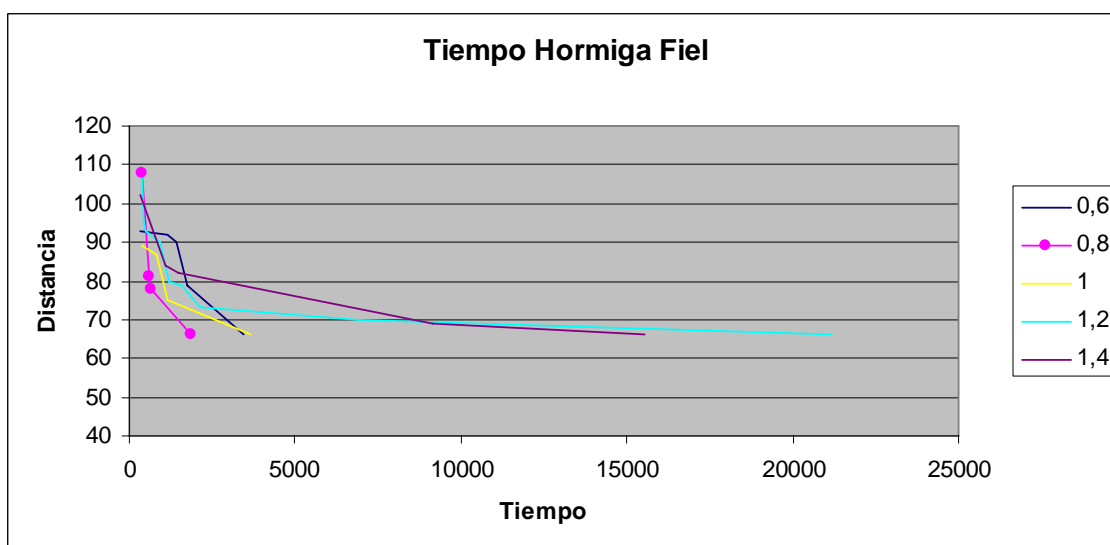


Ilustración 68 – Gráfico. Cálculo del factor de feromona inicial. IV_R1000-FI. Tiempo Hormiga Fiel.

En estos últimos gráficos también se observa lo indicado anteriormente, de como la exploración consigue mejores resultados en la colonia.

○ Conclusión.

El proyecto trata de buscar una distancia aceptable en el menor tiempo posible, la hormiga fiel es la que primero proporciona estos resultados, siendo este el motivo de elegir como valor de factor de feromona el 0,8, para el tamaño de estudio del proyecto.

Factor de feromona inicial = 0,8

3.5.4.4 ANÁLISIS DE DIMENSIONES DE RED.

Con los valores de las variables establecidos, sólo queda realizar las pruebas para calcular hasta que dimensiones es capaz de aportar unos resultados aceptables el algoritmo diseñado.

Estos resultados también servirán para realizar una comparación con los datos obtenidos por el algoritmo inicial, [Punto 3.2.3.](#), y comprobar si este se ha perfeccionado con las modificaciones introducidas para el tipo de problema que estamos tratando de estudiar.

Las pruebas realizadas han tenido tiempo limitado de ejecución. El objetivo del proyecto es dar información aceptable en un espacio de tiempo razonable para el problema en cuestión, así se consideró que las pruebas a realizar no deberían superar un tiempo de 5 minutos de ejecución.

- **Red de 100 puntos. IV_R100-DR**

ENTORNO EXPERIMENTAL
<ul style="list-style-type: none">• Se genera una red de 100 puntos.
<ul style="list-style-type: none">• Cada punto tiene como origen 2, 50 y 100 arcos y como destino de cada arco un punto aleatorio de la red.
<ul style="list-style-type: none">• El valor de feromona inicial es de 1.
<ul style="list-style-type: none">• Número de hormigas de la colonia 100.
<ul style="list-style-type: none">• Porcentaje de hormigas en el origen del 50%.
<ul style="list-style-type: none">• El valor del factor de feromona inicial es de 0,8.
<ul style="list-style-type: none">• Se realizan 3 ejecuciones independientes sobre el mismo entorno.

Tabla 36 – Entorno experimental. Análisis de dimensionamiento de la red. IV_R100-DR.

Se realizan 3 pruebas independientes sobre el mismo entorno, denominadas Pru-1, Pru-2 y Pru-3, mostrando los resultados tanto del ciclo como del tiempo en milisegundos donde ha sucedido el evento. A continuación se muestran la mejor distancia, en ese momento, tanto de la hormiga fiel (*DHF*), como de las hormigas de la colonia (*DHC*). Se marca la celda en verde en aquellos casos donde se ha conseguido el valor óptimo del experimento, valor que viene expresado en la fila inferior de la tabla.



Recorridos de grafos mediante hormigas.

Puntos en la red		100											
Nº Arcos por punto		2				50				100			
		Ciclo	Tiempo	DHF	DHC	Ciclo	Tiempo	DHF	DHC	Ciclo	Tiempo	DHF	DHC
Pru - 1	1 ^{er} Éxito	4	109	27	27	2	47	10	10	2	63	9	9
	Fin	4	109	27	27	135	2828	3	3	98	2235	4	5
Pru - 2	1 ^{er} Éxito	4	109	27	27	3	93	11	11	4	156	19	19
	Fin	4	109	27	27	95	1859	3	5	26	718	4	13
Pru - 3	1 ^{er} Éxito	4	109	27	27	7	234	32	32	6	203	23	23
	Fin	4	109	27	27	120	2609	3	6	288	6797	4	4
Óptimo		27				3				4			

Ilustración 69 – Resultados. Análisis de dimensionamiento de la red. IV_R100-DR.

Para una red de 100 puntos, el éxito es completo en todas las pruebas realizadas, para una complejidad añadida de 2, 50 y 100 arcos. Para los grupos de 50 y 100 arcos, hay que indicar que el tiempo en localizar el óptimo es de 2,5 segundos en el de 50, y 3,5 en el de 100, mientras que para el de 2 arcos es inmediato.

Estos resultados son los más similares en complejidad a las pruebas realizadas en la fase inicial ([resultados](#)). Los resultados son superados de una manera sustancial sin dejar dudas sobre las mejoras aportadas por la implementación del algoritmo.

• **Red de 500 puntos. IV_R500-DR**

ENTORNO EXPERIMENTAL	
•	Se genera una red de 500 puntos.
•	Cada punto tiene como origen 2, 50 y 100 arcos y como destino de cada arco un punto aleatorio de la red.
•	El valor de feromona inicial es de 1.
•	Número de hormigas de la colonia 100.
•	Porcentaje de hormigas en el origen del 50%.
•	El valor del factor de feromona inicial es de 0,8.
•	Se realizan 3 ejecuciones independientes sobre el mismo entorno.

Tabla 37 – Entorno experimental. Análisis de dimensionamiento de la red. IV_R500-DR.

Al igual que en el entorno anterior se realizan 3 pruebas independientes sobre el mismo entorno, denominadas Pru-1, Pru-2 y Pru-3, mostrando los resultados tanto del ciclo como del tiempo en milisegundos donde ha sucedido el evento. A continuación se muestran la mejor distancia, en ese momento, tanto de la hormiga fiel como de las hormigas de la colonia. Se marca la celda en verde en aquellos casos donde se ha conseguido el valor óptimo del experimento, valor que viene expresado en la fila inferior de la tabla.



Recorridos de grafos mediante hormigas.

Puntos en la red		500											
Nº Arcos por punto		2				50				100			
		Ciclo	Tiempo	DHF	DHC	Ciclo	Tiempo	DHF	DHC	Ciclo	Tiempo	DHF	DHC
Pru - 1	1 ^{er} Éxito	9	235	44	44	11	312	50	51	8	196	42	42
	Fin	17	391	32	32	4222	98203	14	23	1006	30240	17	26
Pru - 2	1 ^{er} Éxito	11	265	49	49	10	313	57	57	12	328	64	64
	Fin	58	968	32	32	7241	175719	15	20	1920	70375	18	32
Pru - 3	1 ^{er} Éxito	9	265	46	46	32	703	51	164	6	125	25	25
	Fin	35	656	32	45	1288	43062	8	18	1946	80563	16	17
Óptimo		32				8				7			

Ilustración 70 – Resultados. Análisis de dimensionamiento de la red. IV_R500-DR.

Con 500 puntos el algoritmo empieza a no presentar el valor óptimo. Cuando existen dos arcos por punto no presenta problemas, obteniendo el óptimo en menos de un segundo, con lo que se puede considerar que son unos resultados muy buenos.

Cuando analizamos el resultado con 50 arcos, aparece en uno de los tres intentos el óptimo, pero en las otras dos pruebas no lo consigue. En cuanto a 100 arcos por nodo, los resultados ya no se pueden considerar como aceptables.

• **Red de 1000 puntos. IV_R1000-DR**

ENTORNO EXPERIMENTAL	
•	Se genera una red de 1000 puntos.
•	Cada punto tiene como origen 2, 50 y 100 arcos y como destino de cada arco un punto aleatorio de la red.
•	El valor de feromona inicial es de 1.
•	Número de hormigas de la colonia 100.
•	Porcentaje de hormigas en el origen del 50%.
•	El valor del factor de feromona inicial es de 0,8.
•	Se realizan 3 ejecuciones independientes sobre el mismo entorno.

Tabla 38 – Entorno experimental. Análisis de dimensionamiento de la red. IV_R1000-DR.

También en este entorno se realizan 3 pruebas independientes sobre el mismo entorno, denominadas Pru-1, Pru-2 y Pru-3, mostrando los resultados tanto del ciclo como del tiempo en milisegundos donde ha sucedido el evento. A continuación se muestran la mejor distancia, en ese momento, tanto de la hormiga fiel como de las hormigas de la colonia. Se marca la celda en verde en aquellos casos donde se ha conseguido el valor óptimo del experimento, valor que viene expresado en la fila inferior de la tabla.



Recorridos de grafos mediante hormigas.

Puntos en la red		1000							
Nº Arcos por punto		2				50			
		Ciclo	Tiempo	DHF	DHC	Ciclo	Tiempo	DHF	DHC
Pru - 1	1 ^{er} Éxito	18	406	109	109	33	735	202	202
	Fin	1779	27265	70	70	2319	81891	53	72
Pru - 2	1 ^{er} Éxito	18	360	104	104	37	828	188	188
	Fin	6713	101735	70	75	2776	122562	44	73
Pru - 3	1 ^{er} Éxito	24	438	160	161	17	469	89	89
	Fin	538	8344	70	71	2771	98550	38	71
Óptimo		70				14			

Ilustración 71 – Resultados. Análisis de dimensionamiento de la red. IV_R1000-DR.

Con 500 puntos y 100 arcos los resultados obtenidos ya no eran aceptables. Es esta la causa para eliminar del análisis de este grupo ya la complejidad de 100 arcos por punto.

En este conjunto de prueba se observa que todavía mantiene un éxito completo para dos arcos por nodo, pero para 50 ya presenta unos resultados bastante alejados de su valor óptimo.

• **Red de 2000 puntos. IV_R2000-DR**

ENTORNO EXPERIMENTAL	
•	Se genera una red de 2000 puntos.
•	Cada punto tiene como origen 2, 50 y 100 arcos y como destino de cada arco un punto aleatorio de la red.
•	El valor de feromona inicial es de 1.
•	Número de hormigas de la colonia 100.
•	Porcentaje de hormigas en el origen del 50%.
•	El valor del factor de feromona inicial es de 0,8.
•	Se realizan 3 ejecuciones independientes sobre el mismo entorno.

Tabla 39 – Entorno experimental. Análisis de dimensionamiento de la red. IV_R2000-DR.

En este nuevo modelo sólo se han realizado las pruebas para un número de arcos por punto de 2. También en este entorno se realizan 3 pruebas independientes sobre el mismo entorno, denominadas Pru-1, Pru-2 y Pru-3, mostrando los resultados tanto del ciclo como del tiempo en milisegundos donde ha sucedido el evento. A continuación se muestran la mejor distancia en ese momento tanto de la hormiga fiel como de las hormigas de la colonia. En este conjunto de pruebas ya no se consigue ningún valor óptimo.



Recorridos de grafos mediante hormigas.

Puntos en la red		2000			
Nº Arcos por punto		2			
		Ciclo	Tiempo	DHF	DHC
Pru - 1	1 ^{er} Éxito	45	750	214	214
	Fin	6564	106828	120	150
Pru - 2	1 ^{er} Éxito	38	594	165	188
	Fin	6238	95628	120	128
Pru - 3	1 ^{er} Éxito	331	5015	347	1664
	Fin	10052	172937	119	152
Óptimo		112			

Ilustración 72 – Resultados. Análisis de dimensionamiento de la red. IV_R2000-DR.

Para completar si el éxito se sigue manteniendo con dos arcos por nodo, se comprueba para una red de 2000 puntos. El resultado se puede considerar aceptable ya que se aproxima al óptimo pero sin conseguirlo.

Con estas comprobaciones se consigue formar una idea de las dimensiones a las que es capaz de aportar soluciones el algoritmo implementado, sabiendo que estos se fundamentan en presentar una solución rápida y aceptable, aunque no óptima.

3.5.4.5 EVOLUCIÓN DE LOS RESULTADOS

Se considera oportuno plasmar en el trabajo, como evolucionan las soluciones aportadas por el algoritmo en el tiempo. Se ha elegido una red de una complejidad elevada, en la que conseguir el resultado óptimo fuese complicado y así poder comprobar la evolución de los resultados aportados.

Se han elegido dos redes, de 500 y 1000 puntos, con 10 arcos por nodo. Los resultados obtenidos han sido los siguientes:

ENTORNO EXPERIMENTAL
<ul style="list-style-type: none"> Se genera dos redes de 500 y 1000 puntos.
<ul style="list-style-type: none"> Cada punto tiene como origen 10 arcos y como destino de cada arco un punto aleatorio de la red.
<ul style="list-style-type: none"> El valor de feromona inicial es de 1.
<ul style="list-style-type: none"> Número de hormigas de la colonia 100.
<ul style="list-style-type: none"> Porcentaje de hormigas en el origen del 50%.
<ul style="list-style-type: none"> El valor del factor de feromona inicial es de 0,8.

Tabla 40 – Entorno experimental. Evolución de los resultados. IV_R500-1000-E.

En la tabla inferior se informa de la evolución de los resultados obtenidos en dos ejecuciones independientes sobre una red de 500 y 1000 puntos. El número de arcos por punto es de 10 y se informa del ciclo, tiempo y mejor resultado tanto de la hormiga fiel (DHF) como de la colonia (DHC).

Nº Puntos Nº Arcos	500				1000			
	10				10			
	Ciclo	Tiempo	DHF	DHC	Ciclo	Tiempo	DHF	DHC
	10	156	42	42	19	329	109	109
	37	641	40	40	22	391	82	109
	52	907	39	40	59	1080	69	109
	60	1047	37	40	592	11028	63	66
	106	1861	35	40	1238	23042	49	66
	176	3081	32	35	2456	41469	47	55
	202	3550	30	35	4067	64025	46	55
	216	3800	29	35	4244	66450	42	55
	412	7288	27	30	6266	94263	40	55
	1346	23693	21	27	6786	102365	35	55
	5997	104606	19	25				

Ilustración 73 – Resultados. Evolución de los resultados. IV_R500-1000-E.

Los datos muestran como van mejorando los resultados en el tiempo, hecho importante en los objetivos enunciados del proyecto. Una representación gráfica ayuda para una mejor comprensión:

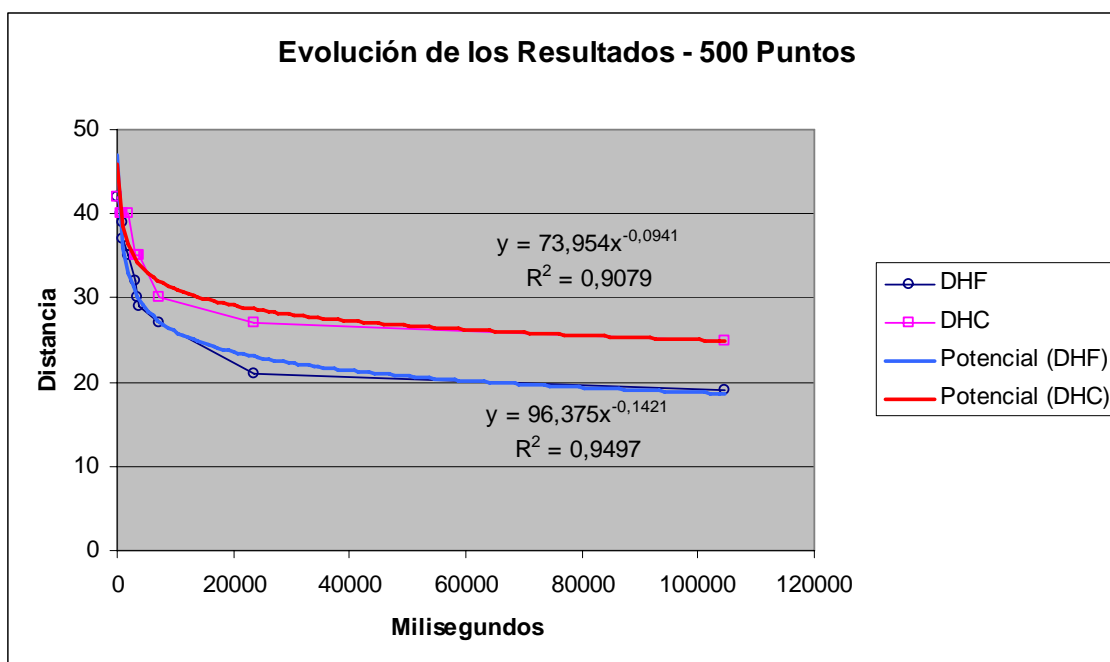


Ilustración 74 – Gráfico. Análisis de evolución de los resultados. IV_R500-E.

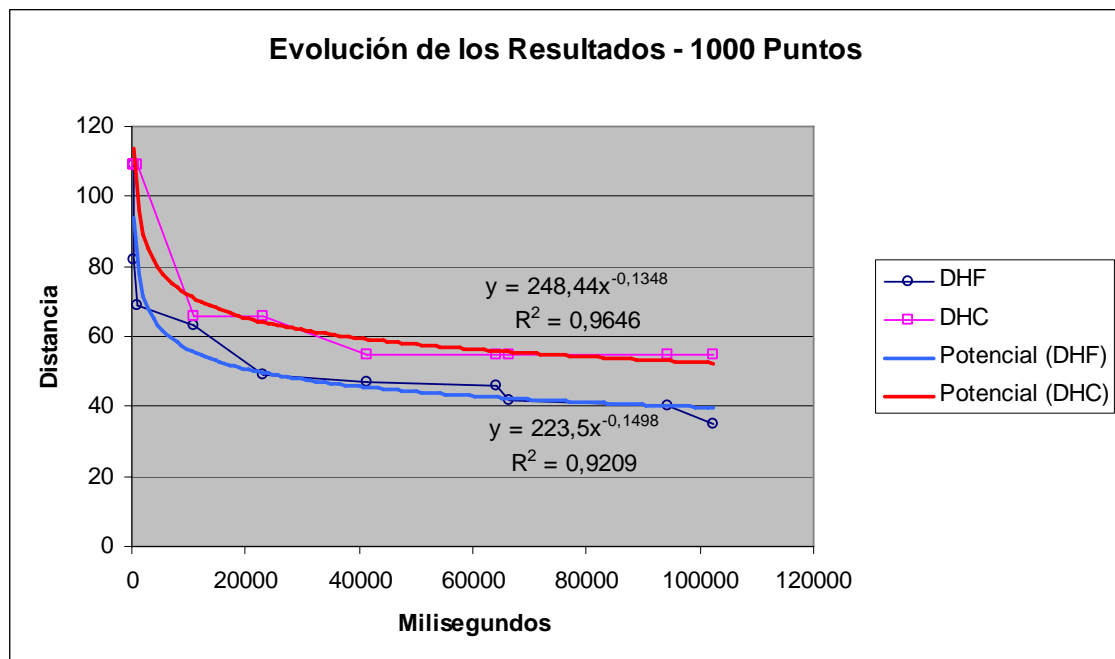


Ilustración 75 – Gráfico. Análisis de evolución de los resultados. IV_R1000-E.

Como se puede observar la evolución de los resultados presenta una similitud a una curva potencial en ambos casos. En un comienzo, la solución realiza un avance rápido, para posteriormente ir decreciendo en el tiempo hasta que llegar a un punto donde la pendiente es inapreciable. En la primera fase, la de mayor pendiente, el algoritmo es donde tiene que presentar su eficiencia para que sea de utilidad. Una rápida respuesta, aunque el resultado no sea el óptimo, es lo que busca este tipo de técnicas.

3.5.5 VERIFICACIÓN DEL AVANCE EN LA OBTENCIÓN DE RESULTADOS POR LA INTRODUCCIÓN DE MEJORAS EN EL ALGORITMO TOMADO COMO BASE.

Los resultados mostrados en el [punto 3.2.3](#), y tomados como punto de partida para la comprobación de los avances efectuados durante la elaboración del proyecto, quedan claramente superados por el algoritmo desarrollado en este último punto.

El algoritmo de partida obtenía unos resultados aceptables con una red de 100 puntos, pero no conseguía alcanzar el valor el óptimo. En cambio, en el algoritmo desarrollado en este punto, la no obtención del óptimo comienza a ocurrir a partir de los 500 puntos de red, llegando incluso a obtener el óptimo con 1000 puntos y 2 arcos por punto.

Con esto queda claramente demostrada una mejora sustancial, en cuanto a efectividad, tomando como referencia el primer algoritmo implementado para la resolución del problema objeto de estudio.



4 CONCLUSIONES Y MEJORAS.

Los resultados obtenidos han demostrado que este tipo de problema puede ser resuelto por algoritmos basados en colonia de hormigas, pero hay que indicar que se ha tenido que realizar una modificación del algoritmo de partida bastante sustancial, incorporando nuevos comportamientos de inspiración biológica que se han considerado interesantes para la consecución de los objetivos.

A pesar de las modificaciones introducidas y los resultados obtenidos, este tipo de algoritmos no se considera adecuado para la resolución de este tipo de problemas, ya que los tiempos transcurridos para proporcionar la primera solución aceptable, pueden considerarse elevados en redes de gran tamaño. En nuestro estudio se han obtenido resultados óptimos en redes de 1000 nodos y 2 arcos por nodo, pero a partir de estas dimensiones, los resultados no pueden considerarse como aceptables en tiempo y proximidad al óptimo.

Basándose en estas dos premisas,

- Para resolver una instancia del problema no se tiene ninguna información de la red objeto de estudio.
- El objetivo es intentar obtener un buen resultado en un corto espacio de tiempo.

Se va a intentar justificar el porque estos algoritmos, basados en colonias de hormigas, no son los más adecuados para resolver este problema.

La justificación también parte de comportamientos naturales de las propias hormigas. Se sabe que las hormigas encuentran un camino mínimo entre el hormiguero y la fuente de alimentación, pero esto lo consiguen cuando tienen identificado, tanto el punto origen como destino, a través de una ruta. En la búsqueda de la fuente de alimentación, que sería la búsqueda del objetivo, las hormigas pueden tardar un tiempo indeterminado, periodo de gran importancia si se desea dar una solución aceptable lo más rápido posible.

Una vez fijado el objetivo y por lo tanto identificada una ruta, la hormiga si es capaz de conseguir una distancia mínima, pero antes tiene que tener identificada esa primera ruta. Cuando tratamos redes de tamaño grande y alta complejidad, nuestras hormigas artificiales se comportan de manera similar, tardan mucho en encontrar el objetivo, hecho incompatible para los resultados que se pretenden obtener.

Otra circunstancia que lleva a considerar que estos algoritmos no son los más apropiados para la resolución del problema, es la existencia de algoritmos deterministas como Dijkstra, que ofrecen unos resultados excelentes, superando sobradamente los ofrecidos por este tipo de algoritmos en la resolución de las instancias del problema tratados en este proyecto. Es el algoritmo que se ha utilizado para la obtención del valor



óptimo en las diferentes pruebas realizadas, aportando los valores de una manera más ágil que los algoritmos usados en las dimensiones de red objeto de estudio.

Todo esto concluye en considerar que este tipo de algoritmos, para la resolución de problemas de caminos mínimos, son útiles para aquellos modelos en los que se cumplen unas características muy concretas donde los algoritmos deterministas no sean capaces de dar una respuesta correcta. Las características que deben presentar estos modelos son las siguientes:

- La existencia de volatilidad en el peso de los arcos o incluso la existencia de estos. La adaptación a variaciones si es compatible con los algoritmos objeto de estudio, en contra de los algoritmos deterministas.
- También pueden ser de utilidad en aquellos modelos donde no sea indispensable una respuesta rápida, y si en cambio exista una búsqueda de camino mínimo. Esto ocurre cuando la comunicación es persistente en el tiempo y se puede ir mejorando el recorrido durante la existencia del proceso.

En realidad, la combinación de los dos modelos anteriores es donde este tipo de algoritmos pueden presentar batalla a los algoritmos deterministas, ofreciendo unos resultados aceptables, ya que los deterministas no serían capaces de proporcionar un resultado correcto.

En cuanto a trabajos futuros a realizar, se considera oportuno mencionar que para la resolución de problemas de establecimiento de una ruta mínima entre dos puntos cualesquiera en redes grandes y con necesidad de respuesta en un corto espacio de tiempo, la solución por el momento se basa en resultados ofrecidos por algoritmos deterministas.

Ahora bien, cuando la red es de un tamaño donde el algoritmo determinista no es capaz de ofrecer una solución por problemas de recursos, la división de la red se ofrece como una posible solución para una respuesta ágil. Se ha pensado en dos modelos que pueden ser validos como objeto de estudio en futuros trabajos:

- **Selección del objetivo:** El poder centrarse en zonas determinadas de la red. Si se disponen de coordenadas de puntos, no hace falta la ejecución del algoritmo sobre toda la red, sino en una zona concreta, comprendida por las coordenadas más una zona considerada de apoyo alrededor de los puntos origen y destino.
- **División del objetivo:** La consideración de diferentes niveles o capas dentro de la red, lanzando el algoritmo sobre zonas muy concretas o sobre la capa correspondiente. Con esto se consigue utilizar el algoritmo determinista originando como resultado el conjunto de soluciones de varias ejecuciones.

También se puede considerar atractivo para un estudio futuro la combinación de técnicas deterministas con modelos de colonia de hormigas. Este modelo sería de



aplicación a redes en las que se pueden realizar variaciones en los pesos de los arcos (actualizaciones de retenciones de tráfico, accidentes, obras, etc). En este caso se aplicaría el algoritmo determinista para determinar la ruta óptima, y esta se mantendría mediante la aplicación de algoritmos basados en colonias de hormigas, que serían capaces de detectar y solventar las incidencias que pudieran surgir.

5 BIBLIOGRAFÍA Y REFERENCIAS

	[ALM00]	Marta Almirón, Enrique Chaparro, Benjamin Barán <u>Optimización basada en sistema de hormigas con heurística de inicialización</u> Universidad Nacional de Asunción-Paraguay
	[ALO00]	Sergio Alonso, Oscar Cordón, Iñaki Fernández, Francisco Herrera <u>La Metaheurística de optimización basada en colonias de hormigas: Modelos y nuevos enfoques</u> Departamento de Ciencias de la Computación e Inteligencia Artificial, Ingeniería Informática- Granada
	[APA07]	<u>A paso de hormiga</u> El mercurio Digital 2007
	[BAC97]	Bäck T., Hammel U. y Schwefel H. <u>Evolutionary Computation: Comments on the History and Current State</u> IEEE Transactions on Evolutionary Computation Estados Unidos, Vol. 1, No. 1, pp. 3-17, 1997.
	[BAR021]	Benjamin Barán, Marta Almirón <u>Colonia de Hormigas en un ambiente paralelo asíncrono</u> Inflyucley 2002. Montevideo Uruguay
	[BAR022]	Lucía Barcos, Victoria M. Rodríguez, M ^a Jesús Álvarez, Francesc Robusté <u>Algoritmo basado en la optimización mediante colonia de hormigas para la resolución del problema del transporte de carga desde varios orígenes a varios destinos</u> V Congreso de Ingeniería del Transporte, Santander 2002
	[BON99]	Eric Bonabeau, Marco Dorigo, Guy Theraulaz Swarm Intelligence 1999 - Oxford University Press
	[BLU03]	Blum, C y Roli, A Metaheuristics in combinatorial optimization: overview and conceptual comparison ACM Computing Surveys, 2003 Vol.35(3), pp 268-308



	[BLU05]	Christian Blum Ant colony optimization: Introduction and recent trends Physics of Life Reviews 2 (2005)
	[BUL97]	Bullnheimer B., Kotsis G. y Strauss C. <u>Parallelization Strategies for the Ant System</u> Reporte Técnico POM 9/97 Universidad de Viena, Viena – Austria, 1997.
	[BUL99]	B. Bullnheimer, R. F. Hartl, y C. Strauss. <u>An improved ant system algorithm for the vehicle routing problem.</u> Annals of Operations Research, 89, pág 319-328, 1999
	[COB05]	Cobo Ortega, Ángel; Serrano Bedia, Ana María, <u>Un algoritmo híbrido basado en colonias de hormigas para la resolución de problemas de distribución en planta orientados a procesos.</u> Universidad de Cantabria. XIII Jornadas de ASEPUMA. 2005;
	[COL92]	Alberto Colomi, Marco Dorigo, Vittorio Maniezzo <u>An Investigation of some properties of an Ant algorithm</u> Proceedings of the parallel problem solving from nature conference Brussels, Belgium, 1992 pag 509-520
	[COY05]	Carlos Alfredo Coy Calixto, Nelson Obregón Neira <u>Implementación en Hidroinformática de un método de optimización matemática basado en colonia de hormigas</u> Trabajo de Grado, Pontificia Universidad Javeriana, Bogota 2005
	[DEN89]	J. -L. Deneubourg, S. Goss <u>Collective patterns and decision making</u> Ethology, Ecology & Evolution Vol. 1, 1989, págs. 295-311.
	[DOR91]	Dorigo M y Maniezzo V, Colomi A. <u>Ant System: An Autocatalytic Optimizing Process</u> Technical Report 91-016, Dipartimento di Eletttronica e Informazione - Politecnico di Milano, Italia, pp. 1-26, 1991.
	[DOR96]	Marco Dorigo, Vittorio Maniezzo, Alberto Colomi; <u>The Ant System: Optimization by a colony of cooperating agents.</u> IEEE Transactions on Systems, Man, and Cybernetics, 1996 Part-B, Vol 26, N° 1 pp 1-13

	[DOR962]	Dorigo M. y Gambardella L., A study of some properties of Ant-Q. IV International Conference on Parallel Problem from Nature Berlin – Alemania: Springer-Verlag, pp. 656-665, 1996.
	[DOR97]	Dorigo M., Gambardella L., Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem IEEE Transactions on Evolutionary Computation, 1997 Estados Unidos, Vol 1, No. 1, pp. 53-66
	[DOR98]	M. Dorigo, G. Di Caro AntNet: Distributed stigmergetic control for communications networks. <i>Journal of Artificial Intelligence Research</i> , 9 Pag. 317-365, 1998.
	[DOR99]	M. Dorigo y G. Di Caro. The Ant Colony Optimization: A new meta-heuristic. D. Corne, M. Dorigo, y F. Glover, editores, New Ideas in Optimization, páginas 11-32. McGraw Hill, London, UK, 1999.
	[DOR03]	M. Dorigo y T. Stützle. The ant colony optimization metaheuristic: Algorithms, applications and advances. F. Glover and G. Kochenberger, editores, Handbook of Metaheuristics, Vol 57, páginas 251-285. Kluwer Academic Publishers, 2003.
	[DOR04]	Marco Dorigo, Thomas Stutzle Ant colony optimization MIT Press, 2004
	[DOR061]	Marco Dorigo, Krzysztof Socha An Introduction to ant colony optimization Technical Report No. TR/IRIDIA/2006-010
	[DOR062]	Marco Dorigo, Gianni Di Caro, Luca M. Gambardella Ant Algorithms for Discrete Optimization Spring 1999, Vol. 5, No. 2, Pages 137-172 Posted Online March 11, 2006.
	[DRE03]	J. Dréo, A. Pétrowski, P. Siarry, E. Taillard Metaheuristics for hard optimization Springer 2003

	[ENG05]	Andries P. Engelbrecht Fundamentals of Computational Swarm Intelligence Wiley, 2005
	[GAM95]	Gambardella L. y Dorigo M., <u>Ant-Q: A reinforcement learning approach to the traveling salesman problem</u> 12th International Conference on Machine Learning. San Francisco, pp. 252-260, 1995.
	[ISA06]	Pedro Isasi Computación con Inspiración biológica Departamento de Informática Universidad Carlos III Madrid - 2006
	[GAM96]	Luca Maria Gambardella, Marco Dorigo <u>Solving Symmetric and Asymmetric TSP's by Ant Colonies</u> IEEE Conference on Evolutionary Computation (ICEC'96), May 20-22, 1996, Nagoya, Japan
	[GAM00]	L. M. Gambardella y M. Dorigo. <u>Ant Colony System hybridized with a new local search for the sequential ordering problem.</u> <i>INFORMS Journal on Computing</i> , 12: 3, páginas 237-255, 2000.
	[GOL89]	Goldberg, D.E. (1989) Genetic Algorithms in Search, Optimization and Machine Learning 1989 - Addison-Wesley
	[GOS89]	Simon. Goss, S. Aron, J. L. Deneubourg, J. M. Pasteels,; <u>Self-organized shortcuts in the Argentine ant</u> <i>Naturwissenschaften</i> , Vol. 76, 1989, págs. 579-581.
	[HOL00]	Diana Holstein <u>Una Metaheurística Co-Evolutiva para el Problema del Viajante de Comercio</u> Trabajo de Grado



Bibliografía y referencias

	[HOL75]	Holland J. H., Adaptation in Natural and Artificial Systems University of Michigan Press, 1975.
	[INF05]	Facultad de Informática <u>Estructura de Datos y Algoritmos - Grafos</u> Universidad Politécnica de Valencia -2005
	[KEN01]	James Kennedy, Russell C. Eberhart, Yuhui Shi Swarm Intelligence: Collective, Adaptive Morgan Kaufmann, 2001
	[LUZ73]	Antonio Luz Furtado Teoría dos grafos Libros Técnicos y Científicos 1973
	[MAN99]	Vittorio Maniezzo, Alberto Colomi, Marco Dorigo <u>The Ant System Applied to the Quadratic Assignment Problem.</u> IEE Transactions on Knowledge and Data Engineering Vol.11 pág.769-778, 1999
	[MEN01]	Benito Mendoza Uso del Sistema de la Colonia de Hormigas para Optimizar Circuitos Lógicos Combinatorios UNIVERSIDAD VERACRUZANA 2001
	[BOT95]	Boticario, J.G., Varios autores. Aspectos Básicos de la Inteligencia Artificial. Editorial Sanz y Torres 1995
	[MOR00]	Rebecca Morelle <u>La sabiduría diminuta</u> http://news.bbc.co.uk/1/hi/spanish/science/newsid_5133000/5133808.stm BBCMundo
	[OSM05]	Osman, I.H, Laporte, G Metaheuristics: A bibliography. Springer Netherlands, 2005 Annals of Operation Research, 63, 513-623



Bibliografía y referencias

	[PAC00]	Julio Paciello, Hector Martinez <u>Optimización multi-objetiva basada en colonias de hormigas. Teoría y estrategias de paralelización.</u> Universidad Nacional de Asunción, San Lorenzo, Paraguay
	[PER00]	M.L.Pérez Delgado <u>Sistemas Multiagente aplicados a la resolución del problema del viajante de comercio</u>
	[RES06]	Mauricio G. C. Resendeand Celso C. Ribeiro <u>Greedy Randomized Adaptive Search Procedures</u> International Series in Operations Research & Management Science Handbook of Metaheuristics 2006
	[RUS04]	Rusell, Stuart J., Norvig, Peter. Inteligencia Artificial: Un enfoque moderno. Pearson Educación, S.A.
	[SCH96]	R. Schoonderwoerd, O. Holland, J. Bruten, L. Rothkrantz. <u>Ant-based load balancing in telecommunications networks</u> <i>Adaptive Behavior</i> , 5: 2, Páginas 169-207, 1996.
	[STU96]	Stützle T. , Hoos H. Improvements on the Ant System: Introducing Max-Min Ant System <i>International Conference on Artificial Neural Networks and Genetic Algorithms</i> , Viena – Austria, 1996.
	[STU00]	Thomas Stutzle, Holger H. Hoos <u>Max-Min Ant System</u> Future Generation Computer Systems Volume 16 pag 889-914 , 2000
	[VAL00]	Víctor Valenzuela <u>Manual de Análisis y Diseño de algoritmos</u> INACAP Informática
	[VAR06]	Varios Autores 5th International Workshop, ANTS 2006 Springer Brussels, Belgium 2006

Guía de instalación y ejecución **A**

A Guía de instalación y ejecución.

A.1 INSTALACIÓN DEL SOFTWARE ACO-UC3M.

La instalación del software desarrollado es bastante sencilla. Dispone de una herramienta de instalación que facilita la realización del proceso. Simplemente la ejecución del Setup.exe, abre una serie de pantallas que guiarán al usuario hasta la completa instalación del software en su equipo.

El programa está desarrollado bajo la tecnología .NET, esta es la razón por lo que para su ejecución se requiere tener instalado el componente NET. Framework.

Las ventanas que guiarán al usuario durante el proceso de instalación son las siguientes:

- I. En esta primera ventana, se informa al usuario de que se va a proceder a la instalación del software en el equipo, así como una advertencia de protección de los derechos de autor.

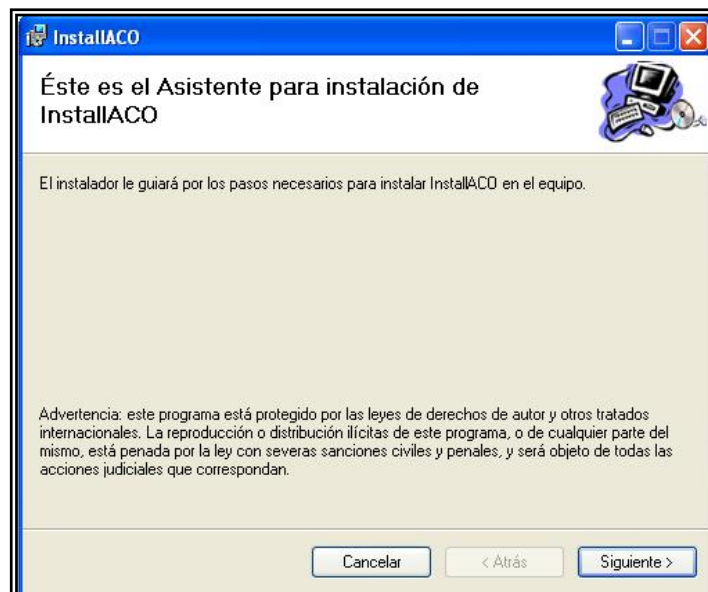


Ilustración 76 – Pantalla de Instalación I.

Pulsando el botón "*Siguiente*", aparecerá una segunda ventana. Si por el contrario, se pulsa "*Cancelar*", aparecerá una confirmación de la cancelación y su aceptación abortará el proceso de instalación.

- II. En esta segunda ventana aparece por defecto la ruta de instalación del software, con la opción de poder modificarla si el usuario así lo desea. Otra opción, que presenta la ventana, es la realización de la instalación para el usuario actual o para todos los usuarios del equipo. Una vez

confirmados estos datos, el usuario tiene la posibilidad de continuar con la instalación pulsando el botón de "*Siguiente*", cancelar el proceso de instalación pulsando el botón de "*Cancelar*", o volver a la pantalla anterior pulsando el botón "*Atrás*".



Ilustración 77 – Pantalla de Instalación II.

III. La siguiente ventana del proceso de instalación informa al usuario de una confirmación de la instalación del software en el equipo. Las opciones son las mismas que la ventana anterior. La pulsación de la opción de "*Siguiente*" confirma la instalación y el inicio del proceso.

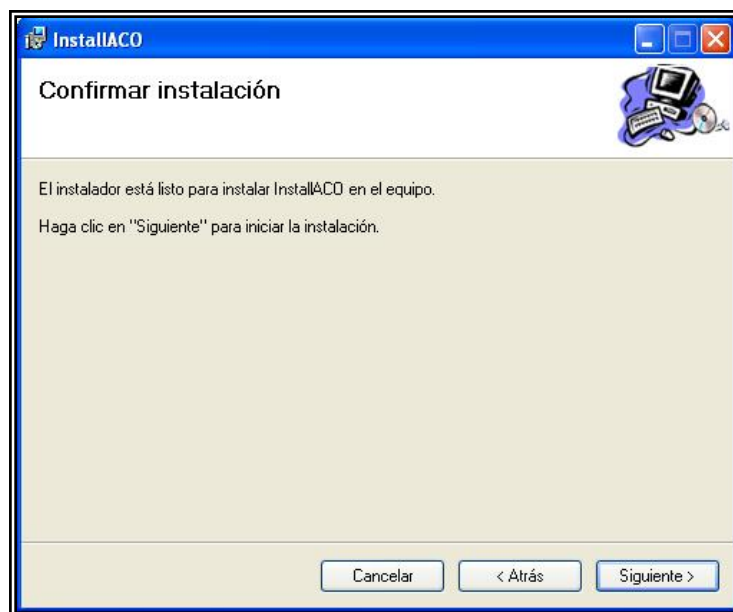


Ilustración 78 – Pantalla de Instalación III.

IV. La última pantalla que aparece, es la confirmación de la completa instalación del software en el equipo, con la única posibilidad de cerrar la ventana con el botón "*Cerrar*", realizando la sugerencia de actualización del software de NET. Framework.

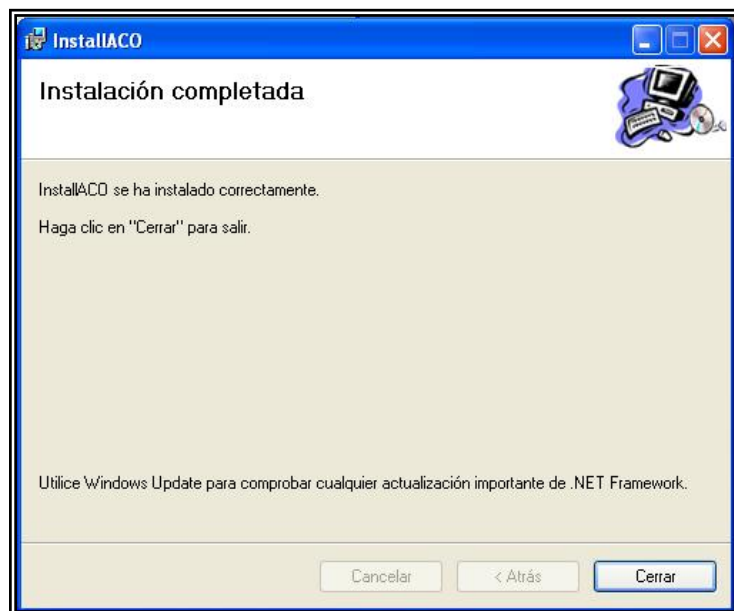


Ilustración 79 – Pantalla de Instalación IV.

Con este último paso, se finaliza la instalación del software ACO-UC3M en el equipo.

A.2 GUÍA DE USUARIO.

El software desarrollado se ha intentado que sea simple de ejecución y de utilización. Consta de una sola pantalla, donde se pueden realizar todas las operaciones exigidas en los requisitos identificados en la fase de análisis.

La instalación del programa creará un acceso directo, en el lugar elegido por el usuario, al fichero “ACO-UC3M.exe”, fichero ejecutable que arrancará el programa para su utilización.

Para una mejor comprensión se van a ir detallando las diferentes funcionalidades, mostrando sus opciones y explicando detenidamente la usabilidad del programa implementado.

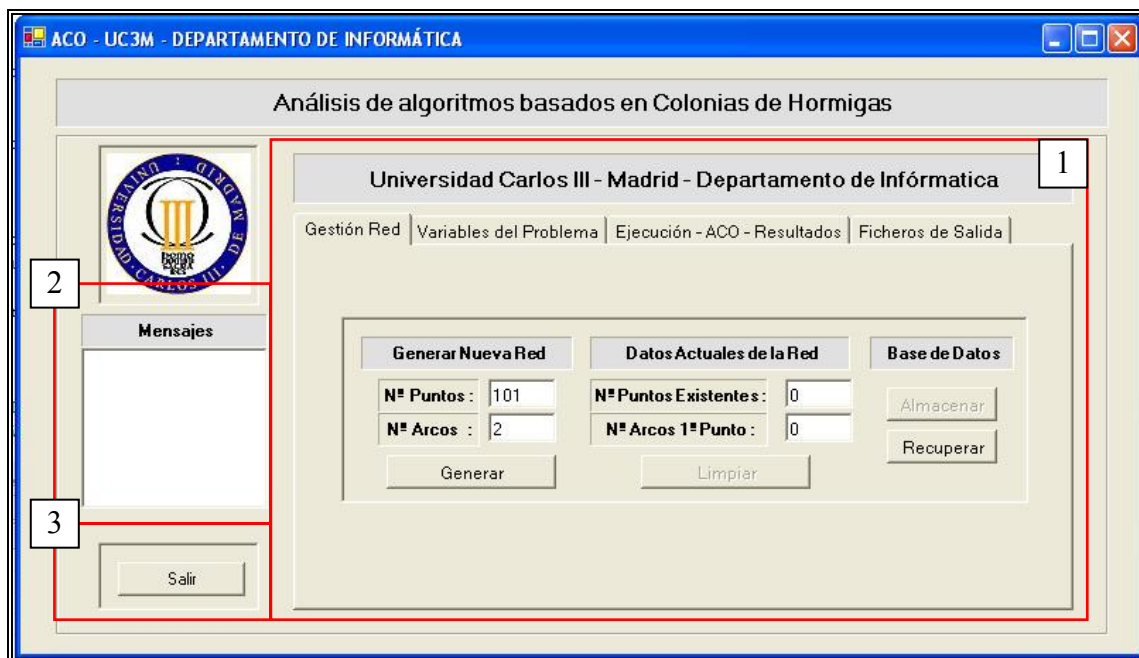


Ilustración 80 – Programa ACO-UC3M- Pantalla de Inicio.

La ventana de ejecución del software, como se observa en la ilustración anterior, presenta tres zonas identificadas con números que a continuación se describen:

1. Zona donde se realiza toda la **gestión de la aplicación**. Se ha considerado oportuno el diseño en carpetas, gestionando cada una de las fases o estados en los que se ha dividido la ejecución de la aplicación.
2. **Cuadro de texto informativo**, donde se presentan los mensajes que se han considerado oportunos durante la ejecución del programa. Es un cuadro de texto con función informativa.
3. Botón “Salir”, que origina la **finalización de la aplicación**.

De las tres zonas marcadas, el punto 1 es el que gestiona en realidad la aplicación. Como consecuencia, pasamos a realizar una explicación más detallada de cada una de las pestañas que lo conforman.

En el programa se han identificado tres fases independientes. Esto ha originado la identificación de tres procesos secuenciales y relacionados, pero claramente independientes. Además, se ha añadido una cuarta pestaña donde el usuario selecciona la información de salida que desea obtener durante la ejecución del software.



A.2.1 GESTIÓN DE RED

La función de esta pestaña es la gestión del grafo sobre el que se realizará la búsqueda de la ruta.

La pestaña presenta cuatro botones y cuatro cuadros de texto agrupados en tres marcos de funcionalidad que a continuación se detallan:



Ilustración 81 – Programa ACO-UC3M- Gestión de Red.

- **Generación Nueva Red:** Presenta dos campos de texto para indicar el número de puntos y arcos que se crearán. Si existe una red ya creada en memoria, se añadirán dichos puntos a la existente.

Una vez introducidos los datos deseados en los campos correspondientes, el botón “Generar”, realizará la acción de creación en memoria de una red con nuevos puntos y arcos de manera completamente aleatoria. El proceso creará el número de puntos indicado, y posteriormente, de cada punto irá creando el número de arcos correspondientes, con un punto destino aleatorio, entre el conjunto de puntos creado.

- **Datos Actuales de la Red:** Existen dos campos informativos que indican tanto el número de puntos actuales, como el número de arcos del primer punto encontrado. Sirven de información del estado actual de la red cargada en memoria.

Existe un botón “Limpiar”, con la función de eliminar de memoria, si la hubiera, la red de puntos y arcos.

- **Base de Datos:** Este apartado consta de dos botones de acción sobre la base de datos. El primero de ellos “Almacenar”, guarda en la base de datos la red que actualmente se encuentra en memoria. El botón “Recuperar”, realiza la función contraria, recuperar de la base de datos la red almacenada. La base de datos se denomina ACO.mdb, y se encuentra en el directorio “datos” de la aplicación.

A.2.2 VARIABLES DEL PROBLEMA.

Esta segunda pestaña muestra las variables que utiliza el algoritmo para la resolución del problema. A continuación se realiza una breve descripción de cada una de ellas:



Ilustración 82 – Programa ACO-UC3M- Variables del problema.

- **Nº Hormigas:** Número de hormigas que forman la colonia para la resolución del problema.
- **En Origen %:** Porcentaje de hormigas que van a ser creadas en el origen de la ruta a resolver. El resto se distribuye aleatoriamente.
- **Feromona Inicio:** Factor de valoración de la feromona de inicio en relación con la depositada. Este valor multiplica a la feromona inicial, causando con su variación dar más o menos significado a la exploración en la resolución del problema.

- **Factor hormiga perdida:** Factor que determina cuando una hormiga es considerada como perdida. El valor significa que cualquier hormiga, cuya distancia recorrida supere tantas veces como indica el valor de la mínima distancia conseguida, sea considerada como perdida.

$\text{Distancia Recorrida} > \text{Mínima distancia conseguida} * \text{Factor de hormiga perdida}$



Hormiga Perdida

- **Punto origen:** Punto de partida de la ruta a resolver.
- **Punto destino:** Punto destino de la ruta a resolver.

A.2.3 EJECUCIÓN Y RESULTADOS.

Esta nueva pestaña permite iniciar el proceso de resolución del problema. También contiene una serie de campos informativos para el usuario, excepto el campo “Ejecución N°”, que permite diferenciar ejecuciones sobre la misma instancia del problema en determinados ficheros de salida.

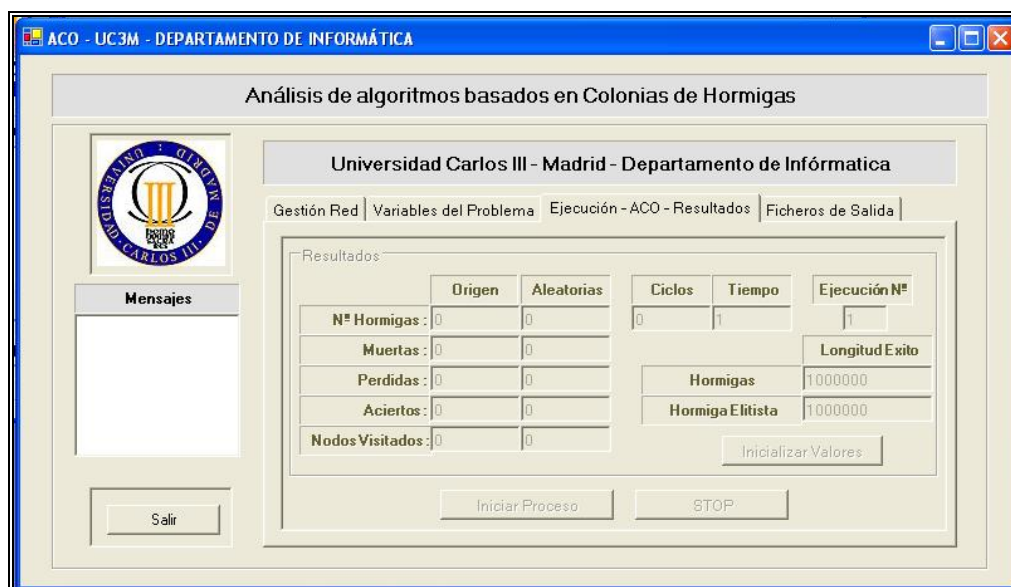


Ilustración 83 – Programa ACO-UC3M. Ejecución-ACO-Resultados.

Presenta los siguientes cuadros de texto informativos:

- **N° hormigas:** Número de hormigas que forman la colonia de exploración de la red
- **Muertas:** Hormigas que llegadas a un punto no pueden desplazarse y no han conseguido su objetivo.



- **Perdidas:** Hormigas que han superado la distancia calculada a partir del factor de hormiga perdida y la mínima longitud hasta el momento conseguida.
- **Aciertos:** Hormigas que han conseguido llegar a su destino.
- **Nodos visitados:** Puntos o nodos de la red visitados.

Todos los campos anteriores hacen referencia a los dos tipos de hormiga generados, tanto a las creadas en origen, como aquellas creadas en lugares aleatorios de la red.

- **Ciclos:** Representa cada turno de movimiento del conjunto de hormigas de la colonia.
- **Tiempo:** Tiempo en milisegundos hasta el momento empleado.
- **Hormigas -Longitud éxito-:** Mínima longitud descubierta hasta el momento por el conjunto de hormigas de la colonia.
- **Hormiga fiel -Longitud éxito-:** Mínima longitud descubierta hasta el momento por la denominada hormiga fiel o de élite.

Además de estos campos, presenta tres botones de acciones que se describen a continuación:

- **Iniciar Proceso:** La pulsación del botón inicia el proceso de exploración de la red por parte de la colonia de hormigas creada.
- **STOP:** Se detiene el proceso de exploración iniciado al pulsar el botón anterior.
- **Inicializar Valores:** Inicializa los valores de información al usuario.

A.2.4 FICHEROS DE SALIDA.

La ejecución puede mostrar los resultados en ficheros de texto si se desea. Existen cuatro ficheros que muestran diferentes conjuntos de datos según las necesidades detectadas durante la elaboración del proyecto. Son casillas de verificación que se activan por el usuario si se desea obtener el fichero correspondiente.

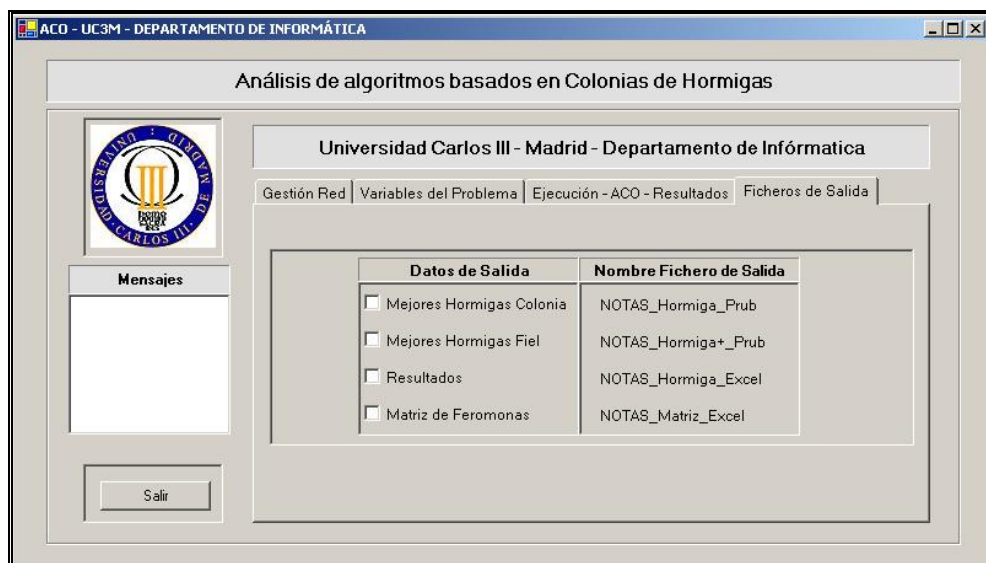


Ilustración 84 – Programa ACO-UC3M- Ficheros de Salida.

Los ficheros presentan la siguiente información:

- **Mejores Hormigas Colonia.** Fichero donde se escribe la información de la hormiga, perteneciente a la colonia creada, que consigue mejorar la distancia mínima entre los puntos origen y destino. Este fichero al generarse presenta la siguiente estructura en su nombre.

NOTAS_Hormiga_Prub_"P"-"NP"-"NA"-"VFI"-"VFP".txt

P - Indicador para que el usuario pueda realizar diferentes pruebas sobre el mismo conjunto y así poder identificar los diferentes resultados.

NP - Número de puntos de la red.

NA - Número de arcos que contiene el primer punto.

VFI - Valor del factor de feromona inicial.

VFP - Valor de factor de hormiga perdida.

La información que proporciona el fichero se escribe en dos líneas:

- En la primera línea se muestra :
 - Posiciones recorridas
 - Tiempo
- En la segunda :



- Ciclo.
- Nodos visitados, Hormigas de situación aleatoria.
- Nodos visitados, Hormigas situadas en el punto inicial.
- **Mejores hormigas fiel:** En este fichero se informa de los datos de la hormiga fiel que consigue mejorar la distancia mínima entre origen y destino. El nombre de este fichero presenta la siguiente estructura:

NOTAS_Hormiga+_Prub_"P"-"NP"-"NA"-"VFI"-"VFP".txt

Los indicadores del nombre del fichero y la información escrita en él son los mismos que los del punto anterior.

- **Resultados:** Este fichero presenta una información reducida de los ficheros anteriores, favoreciendo el tratamiento a posteriori si se desea. La estructura generada es fácil de importar por una hoja de cálculo o base de datos, facilitando cualquier cálculo posterior o elaboración de gráfico, si así se considera oportuno. El nombre del fichero generado es el siguiente:

NOTAS_Hormiga_Resultados.txt

Los datos los escribe en una sola línea con los valores de:

- Ciclo: Ciclo donde se produce la mejora.
- Tiempo: Tiempo transcurrido hasta el momento en milisegundos.
- Distancia hormiga fiel: Nueva distancia mejorada por la hormiga fiel.
- Distancia hormiga colonia: Mejor distancia descubierta por hormigas de la colonia.

En este fichero se escribe una línea cada vez que la hormiga fiel consigue una mejora del recorrido descubierto hasta el momento.

- **Matriz de feromonas:** Este fichero muestra el estado actual de la red en cuanto a la cantidad de feromona depositada en sus arcos. El fichero presenta el siguiente nombre.



NOTAS_Matriz_Resultado.txt

Los datos que se escriben son una línea por cada arco de la red. Se recorren secuencialmente los diferentes puntos y dentro de cada punto los arcos que este contenga, escribiendo las dos cantidades de feromona que presente cada arco. Los campos escritos en el fichero son los siguientes:

- Punto origen. Punto origen del arco.
- Punto destino: Punto destino del arco.
- Feromona Inicial: Cantidad de feromona existente del tipo inicial.
- Feromona Depositada: Cantidad de feromona existente del tipo depositada.

Se ha considerado oportuno no realizar la escritura en cada ciclo, se realiza cada 100 ciclos ya que la ejecución se ralentizaba cuantiosamente.

Ejemplos de todo este conjunto de ficheros se pueden observar en el punto B de este Anexo.

Información de salida **B**



B Información de salida.

B.1 RESULTADOS. ARCHIVOS LOG

En este punto se exponen, a modo de ejemplo, algunos ficheros que han resultado de la ejecución de pruebas realizadas y se detalla el significado de cada campo que se informa en el fichero.

B.1.1 MEJORES HORMIGAS DE LA COLONIA.

NOTAS_Hormiga_Prub-1-1000_10_1_3.txt

Descripción de las variables del nombre del fichero

Prueba 1	Nº Hormigas 1000	Nº Arcos 10
Factor Feromona Inicial 1	Factor Hormiga Perdida 3	

```
**** / 0-0/ 74-7/ 79-15/ 142-16/ 196-22/ 272-27/ 290-32/ 320-36/ 327-44/ 330-53/ 429-60/ 443-63/ 487-
70/ 520-80/ 605-89/ 689-91/ 744-96/ 760-98/ 779-99/ 825-104/ 860-112/ 879-120/ 976-125/ 58-134/ 131-
137/ 159-141/ 217-149/ 231-155/ 325-159/ 377-166/ 421-170/ 510-178/ 581-186/ 600-187/ 646-192/ 722-
196/ 815-202/ 887-207/ 919-209/ 935-211/ 999-214## Distancia :214 ## ***** Tiempo : 567 ms
***** Info ##### Ciclo : 39 ##### Nodos Visitados Aleatorio : 1677 ##### Nodos Visitados Origen : 273
**** / 0-0/ 74-7/ 120-15/ 206-22/ 239-32/ 304-42/ 360-48/ 388-52/ 463-62/ 501-68/ 530-70/ 626-75/ 681-
85/ 715-91/ 770-96/ 786-98/ 809-107/ 817-109/ 819-115/ 839-118/ 891-122/ 900-131/ 976-139/ 999-148##
Distancia :148 ## ***** Tiempo : 5928 ms
***** Info ##### Ciclo : 322 ##### Nodos Visitados Aleatorio : 13879 ##### Nodos Visitados Origen :
2260
**** / 0-0/ 13-8/ 69-16/ 127-24/ 167-32/ 177-39/ 220-48/ 249-50/ 293-54/ 385-59/ 433-65/ 512-73/ 528-
78/ 577-80/ 655-83/ 689-89/ 729-96/ 745-101/ 787-105/ 822-113/ 911-123/ 999-133## Distancia :133 ##
***** Tiempo : 8072 ms
***** Info ##### Ciclo : 445 ##### Nodos Visitados Aleatorio : 19139 ##### Nodos Visitados Origen :
3117
**** / 0-0/ 13-8/ 14-13/ 108-20/ 158-25/ 199-28/ 270-30/ 278-35/ 308-39/ 349-46/ 411-50/ 499-56/ 553-
62/ 595-66/ 671-70/ 767-75/ 794-77/ 828-81/ 833-88/ 883-92/ 889-101/ 946-105/ 999-108## Distancia :108
## ***** Tiempo : 12282 ms
***** Info ##### Ciclo : 687 ##### Nodos Visitados Aleatorio : 29573 ##### Nodos Visitados Origen :
4813
**** / 0-0/ 33-3/ 110-5/ 186-12/ 247-21/ 310-23/ 352-28/ 443-38/ 487-45/ 567-46/ 610-48/ 674-50/ 738-
52/ 829-58/ 854-64/ 951-74/ 983-76/ 999-78## Distancia :78 ## ***** Tiempo : 20875 ms
***** Info ##### Ciclo : 1176 ##### Nodos Visitados Aleatorio : 50572 ##### Nodos Visitados Origen :
8234
**** / 0-0/ 20-6/ 104-11/ 169-14/ 204-16/ 303-26/ 400-31/ 476-38/ 550-43/ 632-45/ 678-48/ 758-51/ 809-
53/ 858-55/ 943-63/ 999-65## Distancia :65 ## ***** Tiempo : 27260 ms
***** Info ##### Ciclo : 1541 ##### Nodos Visitados Aleatorio : 66263 ##### Nodos Visitados Origen :
10788
**** / 0-0/ 29-2/ 98-4/ 129-5/ 147-12/ 240-17/ 295-18/ 375-19/ 437-23/ 494-33/ 565-35/ 660-36/ 670-40/
689-41/ 744-46/ 839-47/ 937-52/ 964-58/ 999-64## Distancia :64 ## ***** Tiempo : 58857 ms
```



Los datos escritos en este fichero corresponden a hormigas pertenecientes a la colonia generada y que han mejorado el resultado hasta el momento detectado.

Cada registro consta de dos líneas, en la primera se proporciona información de la ruta seguida, distancia recorrida y tiempo transcurrido. La segunda línea muestra información general como es el ciclo donde se ha producido el éxito y el número de nodos visitados por cada tipo de hormiga hasta el momento.

B.1.2 MEJORES HORMIGAS FIEL.

NOTAS_Hormiga_Prob+-1-1000_10_1_3.txt

```
**** / 0-0/ 74-7/ 79-15/ 142-16/ 196-22/ 272-27/ 290-32/ 291-41/ 345-45/ 351-51/ 403-58/ 489-65/ 518-67/ 571-69/ 581-73/ 600-74/ 646-79/ 722-83/ 815-89/ 887-94/ 919-96/ 935-98/ 999-101## Distancia :101
## ***** Tiempo : 583 ms
***** Info ##### Ciclo : 40 ##### Nodos Visitados Aleatorio : 1720 ##### Nodos Visitados Origen : 280
**** / 0-0/ 20-6/ 96-13/ 189-18/ 244-19/ 247-25/ 319-31/ 407-35/ 478-43/ 530-47/ 626-52/ 686-56/ 755-62/ 844-69/ 916-72/ 946-75/ 999-76## Distancia :76 ## ***** Tiempo : 6338 ms
***** Info ##### Ciclo : 346 ##### Nodos Visitados Aleatorio : 14878 ##### Nodos Visitados Origen : 2422
**** / 0-0/ 20-6/ 96-13/ 189-18/ 244-19/ 247-25/ 319-31/ 407-35/ 478-43/ 530-47/ 626-52/ 686-56/ 755-62/ 835-65/ 919-67/ 935-69/ 999-72## Distancia :72 ## ***** Tiempo : 20150 ms
***** Info ##### Ciclo : 1135 ##### Nodos Visitados Aleatorio : 48805 ##### Nodos Visitados Origen : 7945
**** / 0-0/ 20-6/ 104-11/ 169-14/ 204-16/ 303-26/ 400-31/ 476-38/ 550-43/ 632-45/ 678-48/ 758-51/ 809-53/ 858-55/ 886-58/ 964-59/ 999-65## Distancia :65 ## ***** Tiempo : 27276 ms
***** Info ##### Ciclo : 1542 ##### Nodos Visitados Aleatorio : 66306 ##### Nodos Visitados Origen : 10794
**** / 0-0/ 20-6/ 104-11/ 169-14/ 204-16/ 303-26/ 400-31/ 476-38/ 550-43/ 632-45/ 678-48/ 758-51/ 809-53/ 821-56/ 899-59/ 943-60/ 999-62## Distancia :62 ## ***** Tiempo : 40221 ms
***** Info ##### Ciclo : 2282 ##### Nodos Visitados Aleatorio : 98126 ##### Nodos Visitados Origen : 15974
**** / 0-0/ 20-6/ 104-11/ 169-14/ 204-16/ 303-26/ 400-31/ 476-38/ 550-43/ 632-45/ 678-48/ 758-51/ 809-53/ 821-56/ 916-57/ 946-60/ 999-61## Distancia :61 ## ***** Tiempo : 66961 ms
***** Info ##### Ciclo : 3805 ##### Nodos Visitados Aleatorio : 163615 ##### Nodos Visitados Origen : 26635
```

Los datos escritos en este fichero, corresponden a la hormiga fiel que ha mejorado el resultado hasta el momento detectado.

Al igual que en el fichero anterior, cada registro consta de dos líneas. En la primera se describe la ruta seguida por la hormiga con los nodos visitados, la distancia recorrida y el tiempo transcurrido hasta el momento. La segunda línea da una información general, como es el ciclo donde se ha producido el éxito y el número de nodos visitados hasta el momento por cada tipo de hormiga.



B.1.3 RESULTADOS.

NOTAS_Hormiga_Resultados.txt

40 583 101 214
346 6338 76 148
1135 20150 72 108
1542 27276 65 65
2282 40221 62 65
3805 66961 61 64

Los resultados que se muestran en este fichero son los mismos que en los ficheros anteriormente detallados, pero su creación tiene como objetivo facilitar el tratamiento de la información que proporciona por hojas de cálculo o base de datos, Los datos van separados de un espacio único, facilitando la importación por este tipo de programas.

Los campos escritos son los siguientes:

Ciclo - Tiempo - Distancia hormiga fiel - Distancia hormigas colonia



B.1.4 MATRIZ DE FEROMONAS

NOTAS_Matriz_Resultados.txt

***** 100 984 *****

- 0 - 5 - 0,1666667 - 0,01515152
- 0 - 99 - 0,1666667 - 0,1666667
- 1 - 42 - 0,02380952 - 0,02380952
- 1 - 70 - 0,02380952 - 0,01639344
- 2 - 47 - 1 - 0
- 2 - 9 - 1 - 0
- 3 - 21 - 1 - 0
- 3 - 44 - 1 - 0
- 4 - 90 - 0,04166667 - 0,04166667
- 4 - 20 - 0,04166667 - 0,0212766
- 5 - 27 - 0,02941176 - 0,02941176
- 5 - 91 - 0,02941176 - 0,01333333
- 6 - 32 - 0,04347826 - 0,04347826
- 6 - 57 - 0,04347826 - 0,03846154
- 7 - 4 - 0,03125 - 0,03125
- 7 - 65 - 0,03125 - 0
- 8 - 74 - 0,009345794 - 0,007352941
- 8 - 68 - 0,009345794 - 0,009345794
- 9 - 84 - 0,02564103 - 0,01960784
- 9 - 12 - 0,02564103 - 0,02564103
- 10 - 90 - 1 - 0
- 10 - 78 - 1 - 0
- 11 - 90 - 0,03571429 - 0,03571429
- 11 - 86 - 0,03571429 - 0,03030303
- 12 - 32 - 0,02702703 - 0,01010101
- 12 - 89 - 0,02702703 - 0,02702703
- 13 - 69 - 1 - 0
- 13 - 33 - 1 - 0
- 14 - 74 - 0,2 - 0,01492537
- 14 - 99 - 0,2 - 0,2
- 15 - 48 - 0,02173913 - 0,02173913
- 15 - 6 - 0,02173913 - 0
- 16 - 16 - 0,01333333 - 0
- 16 - 9 - 0,01333333 - 0,01333333
- 17 - 27 - 1 - 0
- 17 - 81 - 1 - 0
- 18 - 32 - 1 - 0
- 18 - 20 - 1 - 0
- 19 - 32 - 1 - 0
- 19 - 28 - 1 - 0
- 20 - 74 - 0,04347826 - 0,02
- 20 - 31 - 0,04347826 - 0,04347826
- 21 - 10 - 0,01351351 - 0
- 21 - 1 - 0,01351351 - 0,01351351
- 22 - 15 - 0,02702703 - 0,01639344
- 22 - 41 - 0,02702703 - 0,02702703
- 23 - 90 - 1 - 0
- 23 - 49 - 1 - 0
- 24 - 58 - 1 - 0
- 24 - 52 - 1 - 0
- 25 - 95 - 0,02564103 - 0
- 25 - 22 - 0,02564103 - 0,02564103

Ciclo - Tiempo

Origen - Destino - Fer. Inicial - Fer. Depositada



Este fichero va mostrando todos los arcos de la red con la feromona que en ese momento disponen. Cada arco muestra el nodo origen, nodo destino, seguido de los dos valores de feromona, comenzando con la feromona inicial y finalizando con la cantidad de feromona depositada.

C Estructura del CD anexo.

Toda la documentación generada y utilizada en la elaboración del proyecto se ha considerado oportuno reunirla en un CD en forma de anexo.

El CD se estructura en tres carpetas donde se clasificará la documentación generada. Además, se ha considerado conveniente incluir un conjunto de documentos que se han tomado como referencia en la elaboración del proyecto y se encuentran a libre disposición en Internet.

La estructura que presenta el CD es la siguiente:

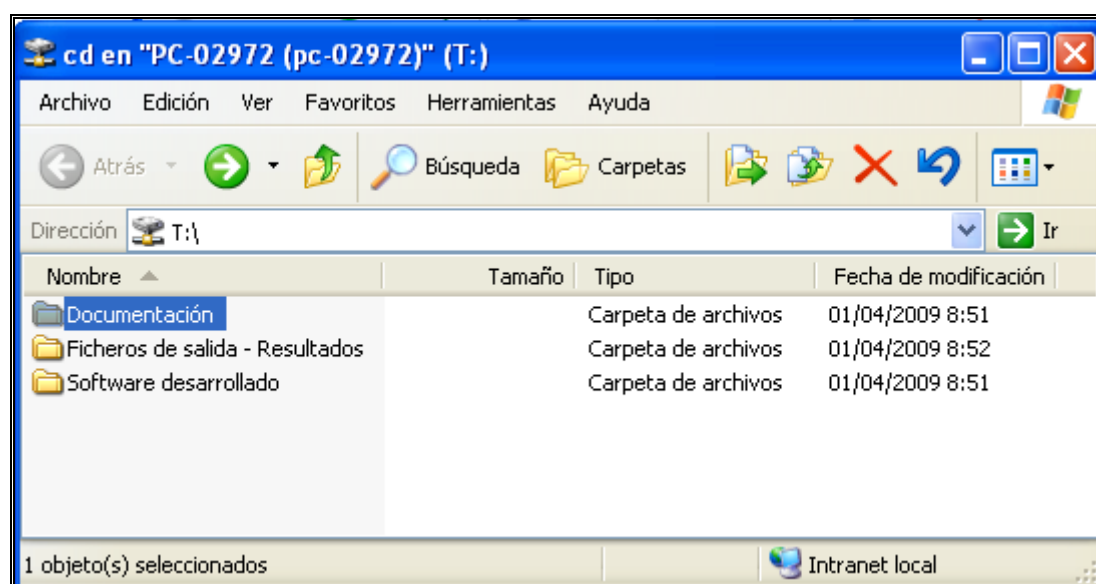


Ilustración 85 – Estructura CD Anexo.

A continuación se realiza una breve descripción de los contenidos de las tres carpetas generadas.

C.1 DOCUMENTACIÓN.

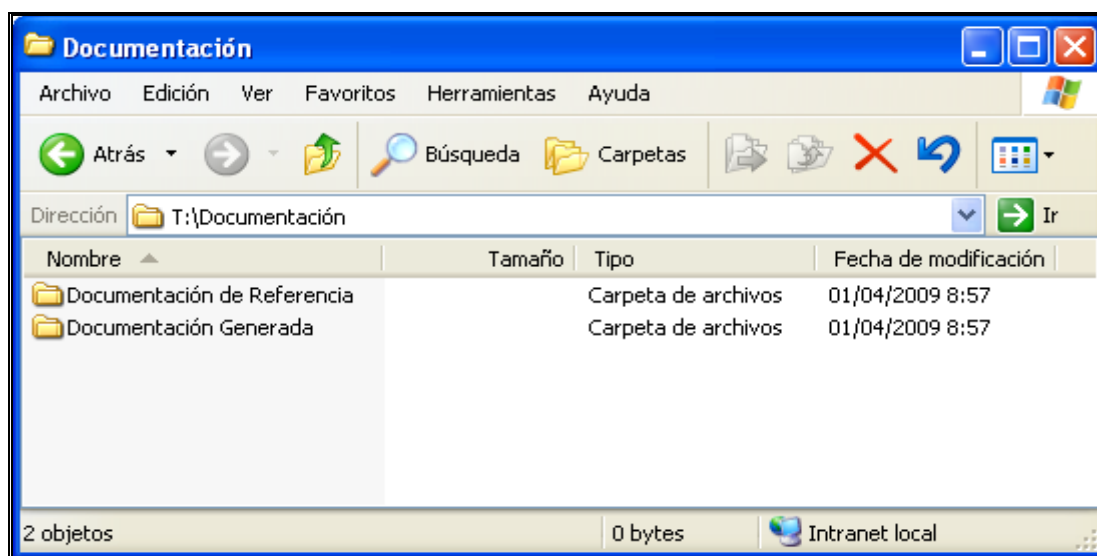


Ilustración 86 – Estructura CD Anexo. Documentación.

Presenta dos carpetas donde se almacenará, por un lado toda la información generada en el proyecto y por otra, textos que han servido de referencia para la elaboración del proyecto.

C.1.1 DOCUMENTACIÓN DE REFERENCIA.

Los documentos se encuentran almacenados en esta carpeta “*Documentación de referencia*”, se identifican por el nombre del documento más la extensión correspondiente. El nombre corresponde al título del documento que aparece en el Capítulo 6 “*Bibliografía y Referencias*”.

Self-organized shortcuts in the Argentine ant.pdf	Simon. Goss, S. Aron, J. L. Deneubourg, J. M. Pasteels. Self-organized shortcuts in the Argentine ant Naturwissenschaft, Vol. 76, 1989, págs. 579-581.
---	---

C.1.2 DOCUMENTACIÓN GENERADA.

En esta carpeta se pueden encontrar, además del presente documento, todos aquellos ficheros realizados en la elaboración del proyecto.

Los ficheros utilizados se centran en tres tipos. En la carpeta “Textos”, se encuentra este fichero tanto en formato Word como pdf. En la carpeta “Presentación”, se encuentra el fichero en formato Powerpoint elaborado para exposición a realizar del proyecto.

La tercera carpeta, denominada “Varios”, es donde se han introducido el resto ficheros desarrollados durante la elaboración del proyecto, ya sean hojas de cálculo (Excel), archivos de diseño (Visio) e Imágenes (jpg).

C.2 FICHEROS DE SALIDA. RESULTADOS.

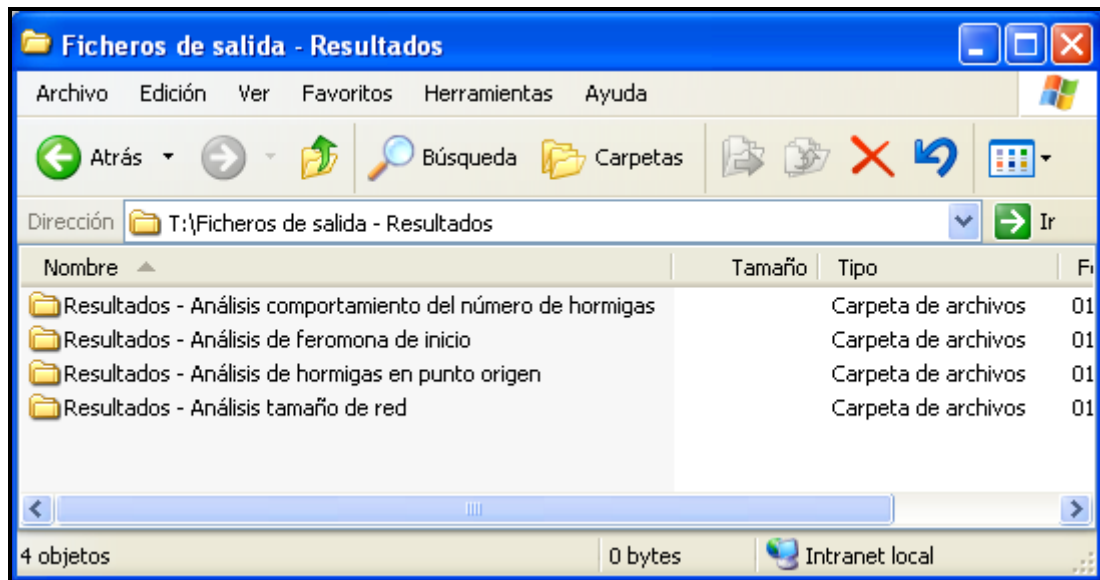


Ilustración 87 – Estructura CD Anexo. Ficheros de salida - Resultados.

En este apartado, se muestra información de los ficheros generados por la aplicación en las diferentes pruebas realizadas y reflejadas en la documentación del proyecto.

C.2.1 RESULTADOS - ANÁLISIS COMPORTAMIENTO DEL NÚMERO DE HORMIGAS.

Un conjunto de pruebas que se consideró importante para la elaboración del proyecto, fue determinar el número de hormigas de la colonia que mejores resultados presentaba en la ejecución del programa, para un tamaño de red similar a los estudios realizados en este proyecto.

Las pruebas presentadas corresponden a tamaños de red de 500, 1000, 2000 y 5000 puntos, mostrando resultados tanto para las hormigas de la colonia, como para la denomina hormiga fiel.

C.2.2 RESULTADOS - ANÁLISIS DE FEROMONA DE INICIO.

Existen dos carpetas que agrupan las pruebas realizadas sobre dos modelos diferentes. Ambas pruebas se realizan sobre una red de 1000 puntos, pero con dos complejidades diferentes, una con 2 arcos por punto y otra con 20.

Estas dos opciones de prueba se realizan sobre diferentes valores de feromona inicial. Este valor varia desde 0,6 hasta 1,4, un rango que se ha considerado adecuado para poder identificar que valor de feromona inicial resulta más provechoso.

C.2.3 RESULTADOS - ANÁLISIS DE HORMIGAS EN PUNTO ORIGEN.

Las diferentes pruebas realizadas de este grupo se han desarrollado sobre dos tamaños de red, una de 500 puntos y otra de 5000. En cada una de ellas se ha comprobado los resultados para un rango de hormigas en origen, que varía de 10% a 60%.

C.2.4 RESULTADOS – EVOLUCIÓN DE ÉXITOS.

Otro punto interesante que se consideró oportuno plasmar en la documentación del proyecto fue comprobar como evolucionaban los resultados en el tiempo, es decir como el algoritmo va consiguiendo encontrar la ruta y mejorándola con el paso del tiempo.

Para ello se eligieron dos ficheros de resultados, generados para redes de 500 y 1000 puntos. Con estos se observó la evolución del algoritmo en el tiempo para la resolución de este tipo de problemas y de esta manera reflejarlo en la documentación.

C.3 SOFTWARE DESARROLLADO.

En esta carpeta se han identificado dos componentes. Por un lado se encuentra el programa para su distribución e instalación y por otro el código desarrollado para la generación del software.

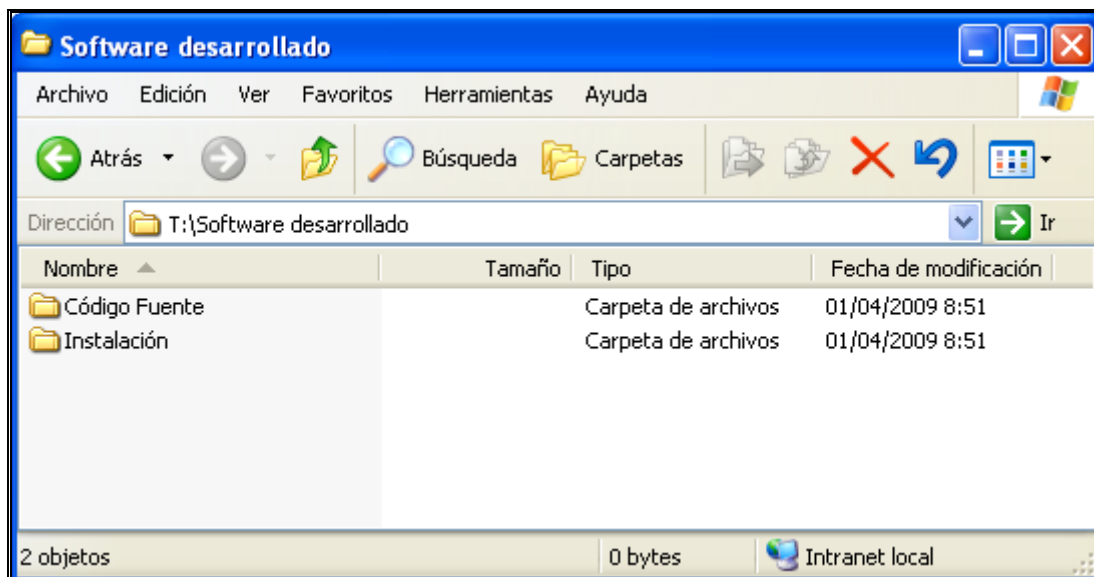


Ilustración 88 – Estructura CD Anexo. Software desarrollado.

C.3.1 CÓDIGO FUENTE.

El software se ha desarrollado en tecnología .NET. Presenta la estructura íntegra de origen de la fase de desarrollo y simplemente ejecutando el fichero del proyecto “ACO-UC3M.vbproj”, cargará todos los elementos necesarios para su comprobación o modificación.



C.3.2 INSTALACIÓN.

En esta carpeta se encuentran los ficheros necesarios para la instalación y ejecución del programa desarrollado. Ejecutando “*Setup.exe*”, se procederá a la instalación del software en el equipo correspondiente.